Practical Network Tomography

THÈSE N^{O} 5332 (2012)

PRÉSENTÉE LE 27 AOÛT 2012

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE POUR LES COMMUNICATIONS INFORMATIQUES ET LEURS APPLICATIONS 3
LABORATOIRE D'ARCHITECTURE DES RÉSEAUX
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Denisa Gabriela GHIŢĂ

acceptée sur proposition du jury:

Prof. E. Telatar, président du jury
Prof. P. Thiran, Prof. A. Argyraki, directeurs de thèse
Prof. A. Krishnamurthy, rapporteur
Prof. J.-Y. Le Boudec, rapporteur
Dr W. Willinger, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2012

Bunicilor mei...

Abstract

In this thesis, we investigate methods for the practical and accurate localization of Internet performance problems. The methods we propose belong to the field of network loss tomography, that is, they infer the loss characteristics of links from end-to-end measurements. The existing versions of the problem of network loss tomography are ill-posed, hence, tomographic algorithms that attempt to solve them resort to making various assumptions, and as these assumptions do not usually hold in practice, the information provided by the algorithms might be inaccurate. We argue, therefore, for tomographic algorithms that work under weak, realistic assumptions.

We first propose an algorithm that infers the loss rates of network links from end-to-end measurements. Inspired by previous work, we design an algorithm that gains initial information about the network by computing the variances of links' loss rates and by using these variances as an indication of the congestion level of links, i.e., the more congested the link, the higher the variance of its loss rate. Its novelty lies in the way it uses this information—to identify and characterize the maximum set of links whose loss rates can be accurately inferred from end-to-end measurements. We show that our algorithm performs significantly better than the existing alternatives, and that this advantage increases with the number of congested links in the network. Furthermore, we validate its performance by using an "Internet tomographer" that runs on a real testbed.

Second, we show that it is feasible to perform network loss tomography in the presence of "link correlations," i.e., when the losses that occur on one link might depend on the losses that occur on other links in the network. More precisely, we formally derive the necessary and sufficient condition under which the probability that each set of links is congested is statistically identifiable from end-to-end measurements even in the presence of link correlations. In doing so, we challenge one of the popular assumptions in network loss tomography, specifically, the assumption that all links are independent. The model we pro-

pose assumes we know which links are most likely to be correlated, but it does not assume any knowledge about the nature or the degree of their correlations. In practice, we consider that all links in the same local area network or the same administrative domain are potentially correlated, because they could be sharing physical links, network equipment, or even management processes.

Finally, we design a practical algorithm that solves "Congestion Probability Inference" even in the presence of link correlations, i.e., it infers the probability that each set of links is congested even when the losses that occur on one link might depend on the losses that occur on other links in the network. We model Congestion Probability Inference as a system of linear equations where each equation corresponds to a set of paths. Because it is infeasible to consider an equation for each set of paths in the network, our algorithm finds the maximum number of linearly independent equations by selecting particular sets of paths based on our theoretical results. On the one hand, the information provided by our algorithm is less than that provided by the existing alternatives that infer either the loss rates or the congestion statuses of links, i.e., we only learn how often each set of links is congested, as opposed to how many packets were lost at each link, or to which particular links were congested when. On the other hand, this information is more useful in practice because our algorithm works under assumptions weaker than those required by the existing alternatives, and we experimentally show that it is accurate under challenging network conditions such as non-stationary network dynamics and sparse topologies.

Keywords:

Network Loss Tomography, Network Measurements, Network Monitoring, Link-loss Inference, Congestion Probability, Correlated Links

Zusammenfassung

In der vorliegenden Doktorarbeit untersuchen wir Methoden für die praxisnahe und genaue Feststellung von Leistungsproblemen in Internetverbindungen. Die vorgeschlagenen Methoden gehören dem Feld der Netzwerkverlusttomographie an, das heisst sie berechnen Verlusteigenschaften von Links mit Hilfe von Messungen zwischen Endpunkten. Die existierenden Versionen von Netzwerkverlusttomographie sind generell unterbestimmt, daher machen die tomographischen Algorithmen zu ihrer Lösung verschiedene Annahmen. Weil diese Annahmen in der Praxis aber nicht immer erfüllt werden, sind die Informationen welche die Algorithmen berechnen teilweise ungenau. Demzufolge plädieren wir für tomographische Algorithmen die mit schwachen, realistischen Annahmen funktionieren.

Zuerst stellen wir einen neuen Algorithmus zur Berechnung von Link-Verlusten vor, das heisst einen Algorithmus der Verlustraten von Netzwerklinks auf Grund von Messungen zwischen Endpunkten berechnet. Inspiriert von existierenden Arbeiten entwickeln wir einen Algorithmus welcher die Anfangsinformationen über das Netzwerk durch das Berechnen von Varianzen von Link-Verlustraten gewinnt: je mehr ein Link überlastet ist, desto höher ist die Varianz seiner Verlustrate. Seine Neuheit liegt in der Art wie diese Informationen verwendet werden - zum Identifizieren und Beschreiben der grössten Menge von Links deren Verlustraten mit Endpunkt-Messungen genau berechnet werden können. Wir zeigen dass unser Algorithmus signifikant höhere Leistungen erbringt als existierende Alternativen, und dass diese Vorteile mit einer steigenden Anzahl von überlasteten Links im Netzwerk zunehmen. Ausserdem überprüfen wir seine Leistung mit einem "Internet-Tomographen" in einer echten Testumgebung.

Zweitens zeigen wir dass Netzwerkverlusttomographie in der Anwesenheit von "Link-Korrelation" angewendet werden kann, das heiss wenn die Verlustrate welche in einem Link auftritt von den Verlustraten von anderen Links im Netzwerk abhängen kann. Wir beweisen dass unter bestimmten wohldefinierten

Bedingungen die Wahrscheinlichkeit, dass jede Menge von Links überlastet ist auf Grund von Endpunkt-Messungen statistisch berechnet werden kann, auch in der Gegenwart von korrelierten Links. Dadurch lösen wir uns von einer verbreiteten Annahme in der Netzwerkverlusttomographie, nämlich von der Annahme dass alle Links im Netzwerk unabhängig sind. Unser Modell nimmt an, dass wir wissen, welche Links wahrscheinlich korreliert sind, aber es verlangt keine Annahme über die Art oder den Grad der Korrelation. In der Praxis behandeln wir alle Links in einem lokalen Netzwerk oder im selben Administrationsbereich als möglicherweise korreliert, denn diese Links können physische Verbindungen, Netzwerkanlagen oder auch Verwaltungsprozesse gemeinsam haben.

Schlussendlich entwickeln wir einen praktikablen Algorithmus zur Berechnung der Überlastungswahrscheinlichkeit welcher die Link-Korrelation berücksichtigt, das heisst er berechnet die Wahrscheinlichkeit dass jede Menge von Links überlastet ist auch wenn die auftretenden Verluste auf einem Link von den Verlusten auf anderen Links im Netzwerk abhängen. Wir modellieren die Berechnung der Überlastungswahrscheinlichkeit als ein System von linearen Gleichungen, so dass jede Gleichung einer Menge von Pfaden entspricht. Weil es nicht praktikabel ist, eine Gleichung für jede Menge von Pfaden im Netzwerk zu berücksichtigen findet unser Algorithmus die grösste Anzahl von linear unabhängigen Gleichungen durch das Auswählen von bestimmten Mengen von Pfaden, basierend auf unseren theoretischen Resultaten. Auf der einen Seite liefert die Berechnung der Überlastungswahrscheinlichkeit weniger Informationen als herkömmliche Alternativen, welche entweder Verlustraten oder Überlastungszustände von Links berechnen, das heisst wir erfahren nur, wie oft eine Menge von Links überlastet ist, aber nicht wie viele Pakete auf einem Link verloren gingen, oder welche Links wann überlastet waren. Auf der anderen Seite ist diese Information in der Praxis nützlicher weil unser Algorithmus unter schwächeren und anspruchsvolleren Annahmen funktioniert als existierende Alternativen, und wir zeigen experimentell dass er bei anspruchsvollen Netzwerkbedingungen wie nicht-stationären Netzwerken und dünnbesetzten Topologien akkurat ist.

Stichwörter:

Netzwerkverlusttomographie, Netzwerkmessung, Netzwerküberwachung, Link-Verlustberechnung, Überlastungswahrscheinlichkeit, Korrelierte Links

Acknowledgments

It has been a great pleasure to have Professors Katerina Argyraki and Patrick Thiran as my thesis advisors. I am grateful for their guidance, trust, and kindness. Katerina's dedication to research inspired me throughout this thesis, and although I was her first student, she proved to be a wonderful advisor right from the start. From Patrick, I have learned among others to strive for clarity when explaining, to simplify proofs, and to make connections across different topics and with real life.

I would like to thank the members of my jury for taking the time to read this thesis, for their insightful and useful feedback, and for a surprisingly enjoyable defense.

A special thank you goes to Professors Tom Henzinger, Willy Zwaenepoel, and George Candea for their support and valuable advice during my doctoral studies.

I am very obliged to the staff of my two labs, LCA3 and NAL, for making everything work so smoothly, and I would like to thank all the people in these labs for creating a friendly and lively work environment.

EPFL is an exciting place with incredibly clever people from all over the world. I consider myself very lucky to have been given the chance to study here and to make many wonderful friends. My PhD life would not have been complete without all the great moments that I have shared with my friends from the doctoral school, the Graduate Student Association (GSA), and the Romanian Student Association (A/RO). They have made the time spent here the best period of my life so far. Thank you! A special thank you goes to Cristi, Iuli, Nicu, and Anuc for the great jokes, scares, and lunches we have shared.

A warm thank you goes to Maria, Lukas, Ghid, and Dan for listening to me, encouraging me, cooking for me, and making Lausanne my home. I am particularly grateful to Maria, who has been like a sister to me and who forced me

into being sociable and open-minded, doing sports, cooking and eating healthy, and having fun. She is partly responsible for who I am today.

I also want to thank Dragoş for his encouragement to start this PhD adventure.

I can never thank enough my family, for their love and support, and for teaching me the importance of education, without which this thesis would not have been possible (Vă mulţumesc mult, mama, tata, şi Alina!).

Last, but not least, I want to thank Lukas for his love and wonderful support, and for making me a better person.

Contents

Abstract **v**

Zusammenfassung **vii**

Acknowledgments **ix**

Contents **xi**

1 Introduction **1**

1.1 Motivation 1

1.2 Outline 3

1.3 Contributions 4

2 Network Loss Tomography **7**

2.1 Network Model 8

2.1.1 Network Topology 8

2.1.2 Path Loss Rates 10

2.2 Continuous Loss Tomography 11

2.2.1 Assumptions 12

2.2.2 Problem Statement 12

2.3 Boolean Loss Tomography 13

2.3.1 Assumptions 13

2.3.2 Problem Statement 14

2.4 State of the Art 14

2.4.1 Multicast-based and Emulated Multicast Methods 15

2.4.2 Unicast-based Boolean Loss Tomography Methods 17

2.4.3 Unicast-based Continuous Loss Tomography Methods . . 19
2.4.4 Non-Tomographic Methods 21
2.5 Common Assumptions in Loss Tomography 21
2.6 Adding the Salt 23
2.7 Conclusions 25

3 Netscope: A Link-Loss Inference Algorithm 27
3.1 The Taming of The Assumptions 28
3.2 The Skyline 29
3.3 Under the Microscope 30
3.3.1 Ordering Links by Loss Rate 31
3.3.2 Finding the Optimal Basis 32
3.3.3 Computing the Loss Rates of Links 32
3.4 Accuracy Analysis 33
3.5 Evaluation 35
3.6 An Internet Tomographer 41
3.6.1 Validation of Link-Loss Inference 42
3.6.2 Loss Characteristics of Internet Links 43
3.7 Conclusions 44

4 Boolean Tomography on Correlated Links 47
4.1 The Forgotten Existence of Correlated Links 48
4.2 Link Correlation Model 52
4.3 Identifiable Link Characteristics 55
4.4 Identification Condition 57
4.5 Congestion Probability 60
4.6 Illustration of Theoretical Results 62
4.6.1 Definitions and Notations 62
4.6.2 The Identifiability++ condition is sufficient. 64
4.6.3 The Identifiability++ condition is necessary. 66
4.7 Theoretical Results 67
4.7.1 A Partial Ordering of Correlation Subsets 67
4.7.2 Some Basic Probabilities 68

4.7.3 Proof of Theorem 4.1 72
4.8 Conclusion 79

5 A Different Loss Tomography 81
5.1 Congestion Probability Inference 82
5.1.1 Assumptions 83
5.1.2 Problem Statement 83
5.2 A Congestion Probability Inference Algorithm 85
5.2.1 Definitions and Notations 86
5.2.2 The System of Equations 89
5.2.3 Theoretical Results 92
5.2.4 The Algorithm 99
5.2.5 Evaluation 110
5.3 A Practical Scenario 117
5.4 Limitations of Boolean Loss Tomography 119
5.4.1 Boolean Loss tomography is ill-posed 119
5.4.2 Analysis of Tomographic Algorithms 120
5.5 Why a Different Loss Tomography? 127
5.6 Conclusion 128

6 Conclusion 131

References 133

A Congestion Probability Inference 137
A.1 A Heuristic to Speed Up the Algorithm 138

B Notations 141

Curriculum Vitae 145

Chapter 1

Introduction

1.1 Motivation

The practical and accurate localization of Internet performance problems is one of the main challenges that network administrators face today. The Internet is a worldwide system of interconnected computer networks that rapidly evolves in an open, unregulated environment. "When something breaks in the Internet, it is hard to figure out what went wrong and even harder to assign responsibility." [1] The difficulty arises from the lack of a central authority in both technological implementation and in establishing policies for access and usage; it is also due to the heterogeneous nature of the Internet. Solving this problem is, nevertheless, crucial in order to guarantee quality of service, verify Service Level Agreements (SLAs), improve network management, enable dynamic routing, and to filter out anomalous or malicious traffic. For example, if the network links that experience excessive loss or delay are known, then real-time applications such as Voice over IP (VoIP) and Video on Demand (VoD) could bypass these links and provide improved user-level performance, or Internet Service Providers (ISPs) would be able to verify if a SLA has been violated and ask for compensation.

In order to localize Internet performance problems, we would ideally need timely and accurate information about the behavior of each network equipment. Suppose that it were possible to collect statistics at each network link. These statistics would include dropped packets rates, delays, connectivity, and available bandwidth. Based on these statistics, we could identify problematic links and pinpoint their particular problems, e.g., an overloaded link that drops pack-

[1] from "Looking Over the Fence at Networks: A Neighbor's View of Networking Research", by Committees on Research Horizons in Networking, National Research Council, 2001.

ets. Unfortunately, this method does not scale to large networks: The collection of such statistics on the Internet imposes a high overhead in terms of traffic delay, computation, communication, and hardware requirements [CCL+04]. Moreover, ISPs regard such statistics as highly confidential and are not willing to make them available to their peers or customers.

Network tomography, however, is an alternative method that provides link statistics without the cooperation of ISPs and with little or no effect on the network load. In essence, network tomography estimates links' characteristics from end-to-end measurements. Specifically, if we can measure the performance of network paths and we know which links are traversed by each path, then we can use network tomography to infer the characteristics of links traversed by the measured paths.

In this thesis, we focus on network loss tomography, where the goal is to infer links' loss characteristics from end-to-end measurements. Current tomographic algorithms attempt to infer either the loss rates of links (i.e., the percentage of packets dropped at each link), or the congestion statuses of links (i.e., infer whether each link is *good* or *congested*, where a link is considered *congested* if it drops more than a certain percentage of the packets it receives). Unfortunately, neither the loss rates, nor the congestion statuses of links are identifiable from end-to-end measurements. Therefore, tomographic algorithms that attempt to infer these quantities must resort to making various assumptions. We discuss and compare these algorithms and their assumptions in Sections 2.4 and 2.5.

Even though network loss tomography is a promising method for determining the loss characteristics of links, in practice, hardly any tomographic algorithm is ever used. The reason for this discrepancy is that state-of-the-art tomographic algorithms cannot assess the accuracy of the information they provide: For most of the assumptions made by these algorithms, there is no way to verify if they hold in a particular network. When the assumptions on which these algorithms rely are not fulfilled, the estimates of the loss characteristics of links may be inaccurate, moreover, there is no way of knowing to what extent they are inaccurate.

In this thesis, we argue for tomographic algorithms that rely on weaker assumptions, verifiable in practice. We believe this is the fundamental property that would make network tomography practical. By providing concrete examples that work in practical scenarios, we show that such algorithms are not confined to the "Island of Utopia."

1.2 Outline

In this thesis, we investigate whether the problem of network loss tomography can be accurately solved under more realistic assumptions than those required by state-of-the-art tomographic algorithms.

First, we formally describe the problem of network loss tomography in Chapter 2. We distinguish two versions of the problem of network loss tomography in the literature: continuous loss tomography, whose goal is to infer the loss rates of links (Section 2.2), and Boolean loss tomography, whose goal is to determine whether each link is good or congested (Section 2.2). We review state-of-the-art algorithms that address these problems in Section 2.4, and we compare them with respect to their assumptions in Table 2.1.

In Chapter 3, we focus on the continuous loss tomography problem, and propose a new link-loss inference algorithm: Netscope. Our algorithm combines first- and second-order moments of end-to-end measurements in a way that significantly outperforms the existing alternatives. We validate Netscope's performance using an "Internet tomographer" that runs on a real testbed, i.e., PlanetLab [Pla].

In Chapter 4, we show that it is feasible to perform network loss tomography in the presence of "link correlations," i.e., when the losses that occur on one link depend on the losses that occur on other links in the network. More precisely, in the presence of link correlations, we formally derive the necessary and sufficient condition under which the probability that each set of links is congested is statistically identifiable from end-to-end measurements.

Finally, in Chapter 5, we design a practical algorithm that solves "Congestion Probability Inference" in the presence of link correlations, i.e., our algorithm infers with which probability each set of links is congested under the link correlation model proposed in Chapter 4. On the one hand, the information provided by our algorithm is less than that provided by the existing alternatives that infer either the loss rates or the congestion statuses of links, i.e., we only learn how often each set of links is congested, as opposed to how many packets were lost at each link, or to which particular links were congested when. On the other hand, this information is more useful in practice, because our algorithm works under assumptions weaker than those required by the existing alternatives, and we experimentally show that it is accurate under challenging network conditions such as non-stationary network dynamics and sparse topologies.

1.3 Contributions

In this thesis, we make the following contributions:

1. We propose Netscope, a new tomographic algorithm that infers the network links' loss rates from end-to-end measurements (Chapter 3). Netscope uses a novel combination of first- and second-order moments of end-to-end measurements to identify and characterize the maximum set of links whose loss rates can be accurately inferred by network tomography. Netscope is robust in the sense that it requires no parameter tuning, moreover, its advantage over the existing alternatives increases with the number of congested links in the network.

   We have built an "Internet tomographer" that runs on PlanetLab nodes and uses Netscope to infer the loss rates of links located between them. We use some of the measured paths for inference and others for validation, and we show that the results are consistent.

2. We show that it is feasible to perform network loss tomography in the presence of "link correlations," i.e., when the losses that occur on one link depend on the losses that occur on other links in the network (Chapter 4). More precisely, we formally derive the necessary and sufficient condition under which the probability that each set of links is congested is statistically identifiable from end-to-end measurements even in the presence of link correlations,. In doing so, we challenge one of the popular assumptions in network loss tomography, specifically, the assumption that all links are independent. Our model assumes we know which links are most likely to be correlated, but it does not assume any knowledge about the nature or the degree of their correlation. In practice, we consider that all links in the same local area network or the same administrative domain are potentially correlated, because they may be sharing physical links, network equipment, or even management processes.

3. We have designed a practical algorithm that solves "Congestion Probability Inference" in the presence of link correlations, i.e., our algorithm infers with which probability each set of links is congested under the link correlation model proposed in Chapter 4 (Chapter 5). We model Congestion Probability Inference as a system of linear equations where each equation corresponds to a set of paths. Because it is infeasible to consider an equation for each set of paths in the network, our algorithm finds the maximum number of linearly independent equations by selecting particular sets of paths based on our theoretical results. Our algorithm works under the weakest set of

assumptions to date, and we experimentally show that it is accurate under challenging network conditions such as non-stationary network dynamics and sparse topologies.

We experimentally show that, in the scenario of an ISP that wants to monitor the performance of its peers, it is more useful to solve Congestion Probability Inference than Boolean loss tomography, because the latter cannot be solved accurately enough in practice. We do not attribute the blame to the limitations of any particular tomographic algorithm, rather to the fundamental difficulty of solving Boolean loss tomography.

Chapter 2

Network Loss Tomography

Network tomography estimates performance parameters based on traffic measurements at a limited number of nodes. Extracting the hidden information from traffic measurements is an inference problem, hence, the term *network tomography* proposed by Vardi [Var96]. Two forms of network tomography have been studied in the literature: origin-destination tomography and link-level inference. Origin-destination tomography estimates path-level parameters from measurements made on individual links; the goal is to determine the intensity of network traffic between all origin-destination pairs, a key input to routing algorithms. The second problem, link-level inference estimates the characteristics of network links from path measurements at a limited number of vantage points. This form of network tomography can be used to collect statistics about network links and pinpoint the problematic ones. Throughout the rest of this thesis, we use the term *network tomography* to refer exclusively to the link-level inference problem.

In this chapter, we describe the network loss tomography problem, i.e., the inference of links' loss characteristics from path measurements, and we discuss the state-of-the-art approaches that address it. The rest of this chapter is organized as follows: We describe the model used by network loss tomography in Section 2.1. We present two versions of the loss tomography problem: continuous loss tomography in Section 2.2 and Boolean loss tomography in Section 2.3. We review related work in Section 2.4, discuss common assumption in loss tomography in Section 2.5, state our viewpoint in Section 2.6, and conclude in Section 2.7.

2.1 Network Model

Network loss tomography takes as input the network topology and the loss rates of paths, and it outputs the loss characteristics of network links. There are various methods to gather the input data: For example, the measurements to obtain the loss rates of paths may be either active (by generating probe traffic) or passive (by monitoring or sampling extant traffic), based on either multicast or unicast traffic, etc. In this thesis, we do not discuss in depth the various methods that can be used to obtain the input data, but we do describe a concrete technique to gather this data in Section 3.6.

In this section, we present the models for the input data. The statistical inverse nature of the network tomography problem and the large number of network links demand the simplest possible models for network traffic that ignore many intricacies of packet transport. The focus is shifted from detailed mathematical modeling of network dynamics to careful handling of traffic measurements, large-scale computations, and model validation [CHRY02].

2.1.1 Network Topology

Links and Paths. We model the network as a directed graph $G = \{V, E\}$, where the set of nodes V represents the network elements, and the set of edges E represents the one-way communication links. A node is either a host that generates and/or receives network traffic, or a router that relays network traffic (an Ethernet switch or an IP router). The set of hosts V^H and the set of routers V^R do not overlap, i.e., $V^H \cap V^R = \emptyset$. Each edge represents a logical link between two network elements. An edge does not necessarily correspond to a physical link; it may represent an IP-level or a domain-level-link – in general, a sequence of physical links between two network elements. The underlying nature of each node and edge depends on the method used for building the network graph. For instance, if an operator relies on traceroute [Jac89] to build the network graph, then each node in the resulting graph represents a layer-3 network element and each edge represents an IP-level link. Throughout this thesis, we use the term *link* to refer to an edge in the network graph.

We define a path as a sequence of links starting from a host and ending at another host. We denote the set of all paths in the network by P. If a path $p_i \in P$ traverses a link $e_j \in E$, then we write $e_j \in p_i$. A path never traverses a link more than once, i.e., there are no routing loops. All links participate in at least one path, i.e., there are no unused links.

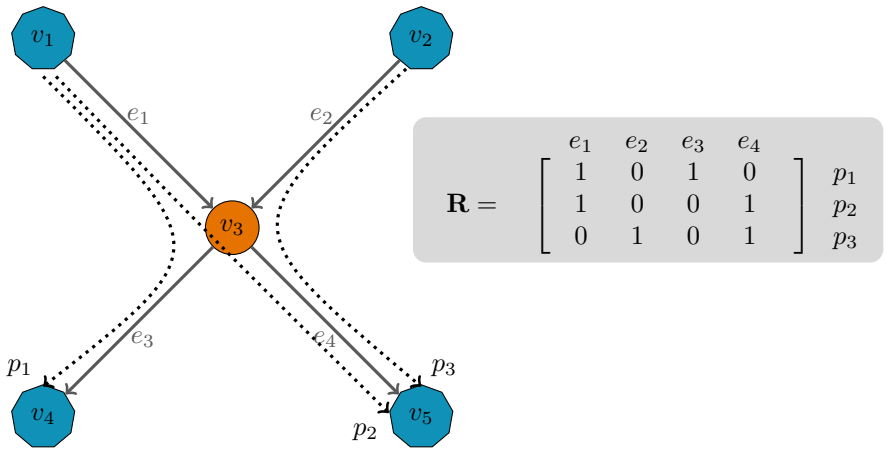


Figure 2.1: A toy topology. Hosts $V^H = \{v_1, v_2, v_4, v_5\}$. Routers $V^R = \{v_3\}$. Links $E = \{e_1, e_2, e_3, e_4\}$. Paths $P = \{p_1, p_2, p_3\}$.

As an example, we consider the toy topology in Figure 2.1. In this topology, node v_3 is a router, while nodes v_1,v_2,v_4, and v_5 are hosts. The set of links is $E = \{e_1, e_2, e_3, e_4\}$, and the set of paths that traverse these links is $P = \{p_1, p_2, p_3\}$.

Routing matrix. Given a network graph $G = \{V, E\}$, and a set of paths P, we compute the routing matrix $\mathbf{R}$ of dimension $|P| \times |E|$, where $|P|$ denotes the number of paths, and $|E|$ denotes the number of links. Each row in the routing matrix $\mathbf{R}$ corresponds to a path in P, while each column corresponds to a link in E. The entry $\mathbf{R}_{i,j} = 1$ if path p_i traverses link e_j, and $\mathbf{R}_{i,j} = 0$ otherwise (The i-th row in $\mathbf{R}$ corresponds to path p_i and the j-th column in $\mathbf{R}$ corresponds to link e_j). In Figure 2.1, we show the routing matrix of our toy topology.

The rank of the routing matrix $\mathbf{R}$ is the number of linearly independent columns of $\mathbf{R}$. We say that $\mathbf{R}$ has *full-column-rank* when all its columns are linearly independent, i.e., $Rank(\mathbf{R}) = |E|$, and we say that $\mathbf{R}$ is *rank-deficient* when $Rank(\mathbf{R}) < |E|$.

A fundamental assumption required by network tomography is that the routing matrix is stable.

Assumption 1. ***Routing Stability:*** *The routing matrix does not change throughout the measurement period.*

This assumption is violated when there are routing changes during the measurement period. In order to minimize the error introduced by the fact that the

routing matrix changes, we must measure the network topology frequently such that we detect when a path has changed, and we discard it from our measurements.

From end-to-end measurements, we cannot distinguish between links traversed by the exact same paths. When such links are consecutive, we can merge them into one single logical link [NT07a]. All our network topologies are pre-processed in this manner. When such links are not consecutive, network tomography cannot identify their performance characteristics. The condition that no two links are traversed by the exact same paths is equivalent to the requirement that all columns in the routing matrix are distinct.

Assumption 2. ***Link Identifiability:*** *All columns in the routing matrix are distinct.*

All tomographic algorithms require as input the network topology in the form of the routing matrix $\mathbf{R}$, for which the Routing Stability and the Link Identifiability assumptions hold.

Input 1. ***The Routing Matrix:*** *The network topology represented by the routing matrix* $\boldsymbol{R}$.

2.1.2 Path Loss Rates

In addition to the network topology, loss tomography also requires as input the loss rates of all paths in P.

We divide time into even slots called *snapshots*, such that an experiment consists of N consecutive snapshots. We model the transmission rate of path p_i during the n-th snapshot with the random variable $\widehat{\phi}_{p_i}(n)$, which is the fraction of packets that are delivered correctly to their destination out of all packets sent on p_i during the n-th snapshot, with $n = 1, \ldots, N$. We model the loss rate of path p_i during the n-th snapshot with the random variable $1-\widehat{\phi}_{p_i}(n)$. Similarly, we model the transmission rate of link e_j during a snapshot with the random variable $\widehat{\phi}_{e_j}(n)$, which is the fraction of packets that are delivered correctly to their next link out of all packets send on e_j during that snapshot. We model the loss rate of link e_j during the n-th snapshot with the random variable $1-\widehat{\phi}_{e_j}(n)$.

Most tomographic algorithms require as input the loss rate of each path in P during a single snapshot, i.e., $N = 1$. Throughout this thesis, we refer to such algorithms as *single-snapshot algorithms*. Recent work [NT07a, NT07b] has shown that we can gain additional information about the network, e.g., the

probability that each link is congested or the variances of the loss rate of links, if we consider measurements over multiple consecutive snapshots, i.e., $N > 1$. Throughout this thesis, we refer to such algorithms as *multiple-snapshot algorithms*. On one hand, the additional information allows multiple-snapshot algorithms to bypass two important assumptions generally made by single-snapshot algorithms: (i) all links are equally likely to be congested, and (ii) the number of congested links is small. On the other hand, in order to be able to gather measurements over multiple-snapshots, the characteristics of links must remain stable for a longer period of time. Hence, multiple-snapshot algorithms need to make an additional assumption compared to single-snapshot algorithms.

Assumption 3. ***Stationarity:*** *For any link e_j, the random variables $\widehat{\phi}_{e_j}(n)$, $n = 1, \ldots, N$, are identically distributed.*

Thus, the behavior of each path and each link can be modeled as a stationary random process. Given a path p_i, since all random variables $\widehat{\phi}_{p_i}(n)$, with $n = 1, \ldots, N$, are identically distributed, we denote by $\widehat{\phi}_{p_i}$ the transmission rate of path p_i during any snapshot. Similarly, given a link e_j, we denote by $\widehat{\phi}_{e_j}$ the transmission rate of link e_j during any snapshot.

For both single-snapshot and multiple-snapshot algorithms, we formally describe the path loss rates required as input.

Input 2. ***Path Loss Rates:*** *The loss rates of all paths represented by the random variables $1 - \widehat{\phi}_{p_i}$, with $p_i \in P$, in each snapshot.*

In conclusion, all tomographic algorithms require as input the network topology represented by the routing matrix $\mathbf{R}$ and the path loss rates represented by the random variables $1 - \widehat{\phi}_{p_i}$, for all $p_i \in P$, in each snapshot.

2.2 Continuous Loss Tomography

Continuous loss tomography infers the loss rates of network links in a given snapshot, i.e., it determines the value of the random variables $\widehat{\phi}_{e_j}$, for all $e_j \in E$, in that particular snapshot. The gist behind continuous loss tomography is to establish a linear relationship between the loss rate of a path and the loss rates of all links traversed by that path. Towards this goal, it needs to make certain assumptions about the loss rates of network links.

2.2.1 Assumptions

In order to be able to establish linear relationships between the loss rates of paths and those of links, continuous loss tomography makes two assumptions.

Assumption 4. ***Link Independence:*** *The transmission rates of links, i.e., the random variables $\widehat{\phi}_{e_j}$, for all $e_j \in E$, are independent.*

This assumption implies that the losses that occur on a link are independent from the losses that occur on any other link in the network. The Link Independence assumption, which is justified by earlier work [Duf06, PQW03], considerably simplifies the network tomography problem as later discussed in Section 4.2.

Assumption 5. ***Loss Uniformity:*** *The fraction of packets lost on link is the same for all paths traversing that link.*

This assumption requires that paths experience the same performance degradation on the shared links. In order for it to hold, the number of probe packets sent by active measurements or the number of packets sampled by passive measurements must be large enough. The loss uniformity assumption is justified in [NT07b].

2.2.2 Problem Statement

If the Link Independence and the Loss Uniformity assumptions hold, the transmission rate of path p_i is given by the product of the transmission rates of all links traversed by path p_i,

$$\widehat{\phi}_{p_i} = \prod_{e_j \in p_i} \widehat{\phi}_{e_j}.$$

If we take the logarithm of the above equation, we get a linear equation, i.e.,

$$\log \widehat{\phi}_{p_i} = \sum_{e_j \in p_i} \log \widehat{\phi}_{e_j}. \tag{2.1}$$

Let $Y_{p_i} = \log \widehat{\phi}_{p_i}$, for all paths $p_i \in P$, and $X_{e_j} = \log \widehat{\phi}_{e_j}$, for all links $e_j \in E$. We group these variables in two vectors: $\mathbf{Y} = [Y_{p_i}]_{p_i \in P}$, and $\mathbf{X} = [X_{e_j}]_{e_j \in E}$, and based on Equation 2.1, we form the system of linear equations:

$$\mathbf{Y} = \mathbf{R} \cdot \mathbf{X}, \tag{2.2}$$

where $\mathbf{Y}$ is the vector of available measurements, $\mathbf{X}$ is the vector of unknowns, and $\mathbf{R}$ is the routing matrix. Because the routing matrix $\mathbf{R}$ is always rank-deficient as shown in [NGK+09], the system in Equation 2.2 is undetermined, and we need additional information to identify the vector of unknowns $\mathbf{X}$, and subsequently, the loss rate $1 - \widehat{\phi}_{e_j}$ of each link $e_j \in E$. Link-loss inference algorithms differ in the way they gather this additional information and the assumptions they make in order to solve Equation 2.2.

2.3 Boolean Loss Tomography

The emergence of Boolean loss tomography is motivated by the observation that usually there is only one link on a path that is responsible for the majority of the losses on the path, and in many cases it suffices to know the locations of these links. Unlike continuous loss tomography, which determines the loss rates of links during a snapshot, Boolean loss tomography aims for the simpler goal of identifying if the loss rates of links exceed a certain threshold. We say that a link e_j is *congested* if its transmission rate $\widehat{\phi}_{e_j}$ is below a link-congestion threshold t_l. Similarly, we say that a path p_i is *congested* if its transmission rate $\widehat{\phi}_{p_i}$ is below a path-congestion threshold t_p. A link or a path that is not congested is considered *good*. The goal of Boolean loss tomography is to establish a relationship between the congestion status of a path, and the congestion statuses of all links traversed by that path. Boolean loss tomography was first stated in [Duf06], where the authors discussed the necessary assumptions, and described the problem statement.

2.3.1 Assumptions

In order to establish a relationship between the congestion status of a path and those of links traversed by that path, Boolean loss tomography makes the following assumption.

Assumption 6. ***Separability:*** *A path is good if and only if all the links it traverses are good.*

Consequently, a path is congested if at least one of the links it traverses is congested. This assumption is closely related to the problem of setting the link-congestion threshold t_l, and the path-congestion threshold t_p. Previous work [Duf06] argues for a path-congestion threshold t_p that depends on the

number of links d on each path, i.e., $t_p = 1 - (1 - t_l)^d$ and proposes a value of $t_l = 0.01$.

2.3.2 Problem Statement

In order to formally describe Boolean loss tomography, we introduce two definitions:

Definition 2.0.1. *The random variable Z_{e_j} is the indicator of the congestion status of link e_j during a snapshot, i.e.,*

$$Z_{e_j} = \begin{cases} 1, & \text{if link } e_j \text{ is congested in that snapshot} \\ 0, & \text{otherwise.} \end{cases}$$

Definition 2.0.2. *The random variable W_{p_i} is the indicator of the congestion status of path p_i during a snapshot, i.e.,*

$$W_{p_i} = \begin{cases} 1, & \text{if path } p_i \text{ is congested in that snapshot} \\ 0, & \text{otherwise.} \end{cases}$$

If the Separability assumption holds, we can establish a relationship in Boolean algebra between the congestion status of a path and the congestion statuses of all links traversed by that path:

$$W_{p_i} = \bigvee_{e_j \in E} \mathbf{R}_{i,j} \cdot Z_{e_j}, \text{ for all } p_i \in P. \tag{2.3}$$

Unfortunately, also in the case of Boolean loss tomography, the congestion statuses of links are generally not uniquely identifiable (see Section 5.4.1), hence, Boolean tomographic algorithms make various assumptions in order to find the congested links.

2.4 State of the Art

The term of *network tomography* was proposed by Vardi [Var96], the first to rigorously study the problem of estimating origin-destination traffic intensities from link measurements. The dual problem, link-level inference, arose as a consequence of the fact that, in large-scale networks, we cannot rely on the network to cooperate in characterizing its own behavior.

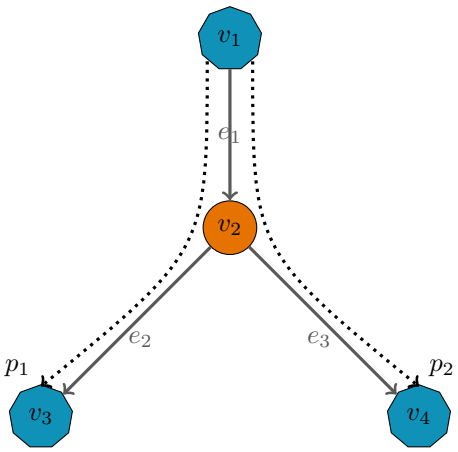


Figure 2.2: The simplest tree topology. Hosts $V^H = \{v_1, v_3, v_4\}$. Routers $V^R = \{v_2\}$. Links $E = \{e_1, e_2, e_3\}$. Paths $P = \{p_1, p_2\}$.

2.4.1 Multicast-based and Emulated Multicast Methods

The first approaches to the loss tomography problem appeared in the context of multicast traffic. The Multicast-based Inference of Network-internal Characteristics (MINC) Project [min] stimulated much work that relies on multicast probes to infer the characteristics of links [CDHT99, DPPT01, BDPT02, DHT$^+$, ABF$^+$00]. The gist is that multicast probes introduce correlation in the end-to-end losses measured by receivers. This correlation can be used to infer the loss characteristics of links in the network. Consider the network depicted in Figure 2.2. If a multicast probe is sent by node v_1 to both receivers, node v_3 and node v_4, but the probe arrives only at node v_4, and not at node v_3, then we can immediately infer that the loss occurred on link e_2. This is because successful reception at host v_4 implies that the multicast probe was forwarded by node v_2. By sending many multicast probes from source v_1 to receivers v_3 and v_4, we can infer the loss rates on the two links e_2 and e_3. Furthermore, using the system in Equation 2.2 for the network in Fig. 2.2, we can infer the loss rate on link e_1. One of the pioneering approaches that fully analyzed this method for a multicast tree is described in [CDHT99]. The authors developed *maximum likelihood estimators (MLEs)* for the loss rates of links, and showed that the inferred link loss rates converge to the true loss rates as the number of multicast probes increases. This approach proposed for multicast trees is further extended to a collection of trees in [BDPT02]. The authors of [BDPT02] establish necessary and sufficient conditions for the identifiability of link loss rates from end-to-

end multicast measurements, and propose two algorithms that estimate the loss rates for a set of links of interest.

Nevertheless, most networks do not support multicast due to scalability limitations since the routers need to maintain state for each multicast group. Furthermore, routers treat multicast traffic differently than unicast traffic, and since most traffic is unicast, concerns arise about the accuracy of the estimates of the loss rates of links. Hence, new tomographic methods emerged that emulated multicast probes with trains of back-to-back unicast probes [CN00, DPPT01]. The authors of [CN00] propose a measurement technique based on losses experienced by unicast back-to-back probe pairs. The motivation of their approach is that if two back-to-back packets are sent on a link, and the first packet successfully traverses the link, then it is highly likely that the second packet will also traverse the link successfully. Consider a node v_k in a multicast tree, and denote by β_k the conditional probability that the second packet of a pair arrives at node v_k, given that the first packet of the pair arrived successfully at node v_k. The authors argue that the conditional probabilities β_k, with $v_k \in V$, are close to 1. Consequently, by sending two back-to-back packets from the same source to two different receivers, they can exploit the correlation between packet-pair losses on common subpaths. Based on this measurement scheme, the authors develop algorithms for likelihood analysis and estimation of the link loss rates and of the conditional probabilities β_k, $k = 1..|V|$. This approach was later extended in [TCN01]. However, estimating both the link loss rates and the conditional probabilities β_k is a very hard problem, and when one or more of the conditional probabilities β_k are less than one, then a systematic bias is introduced into the estimation process and the maximum likelihood estimators are not consistent. For example, in the two-leaf tree in Fig. 2.2, if $\beta_1 < 1$, then the approach in [CN00] overestimates the loss rate on link e_1, and underestimates the loss rates on links e_2 and e_3.

The method proposed by [DPPT01] resembles closely the one described in [CN00]. It proposes a measurement procedure based on stripes, composite probes of unicast back-to-back packets whose collective statistical properties closely resemble those of a multicast packet. The difference from [CN00] is that stripes which consist of several packets are used as opposed to pairs of packets. The authors argue that this results in significantly higher correlation, and that the length and the width of a stripe can be adjusted such that the conditional probabilities β_k are close to 1. Furthermore, they extend the estimator developed for multicast trees [CDHT99] to comply with the striped unicast probing technique and show that the bias introduced by imperfect correlations can be

compensated by using wider stripes if the coalescence property holds, i.e., successful transmission of a packet in the stripe becomes more likely when more other packets from the stripe have been successfully transmitted.

Emulated multicast approaches remove the need for multicast deployment at the cost of somehow lower accuracy and higher administrative costs. The challenge is that the packets in a stripe must have identical experiences when traversing common portions on their paths to their destination. However, congestion events may not affect all packets in a stripe uniformly. For example, correlations within stripes is not perfect when the duration of the congestion event is narrower than the temporal width of the stripe, or when the congestion event starts or stops during the transmission of the stripe. Furthermore, packets in a stripe may become uncorrelated because of packet-dropping in the routers on the basis on Random Early Detection (RED). In addition, the authors of [SB07] show that crafted probe streams can be problematic on testbeds deployed across the Internet, e.g., PlanetLab [Pla]. The host systems in such testbeds are usually overloaded, which can significantly alter the temporal spacing between packets. Therefore, custom software as the one developed by [SB07] must be used in order to guarantee stripe timings. Last, but not least, the algorithms used to infer the link loss rates in these approaches are computationally expensive for real-time applications in large networks. For a general overview of the tomographic methods based on multicast traffic, or on unicast traffic that emulates multicast, we refer the reader to [CHRY02, CCL+04].

2.4.2 Unicast-based Boolean Loss Tomography Methods

The shortcomings of tomographic methods based on multicast or emulated multicast traffic, motivated the development of inference algorithms that can work with readily available measurements. The key idea is that even if the probes are not temporally correlated, it is still expected that two probe streams which traverse a given link over the same period of time, exhibit some correlations in performance. But without strong temporal correlations between probe packets, the link loss rates are not statistically identifiable from end-to-end measurements [Duf06]. Thus, the methods that followed [PQW03, Duf06, BMT05, NT07a] considered the simpler goal of identifying the congested links, i.e., identifying if the link loss rates exceed some threshold, instead of computing their actual values. These approaches solve the Boolean network tomography problem stated in Equation 2.3 by making additional assumptions, one of the most common being that there are only few congested links in the network.

The pioneering work in Boolean tomography was proposed by [PQW03]. The authors developed three techniques to infer the congested links in the network: Random Sampling, Linear Optimization, and Bayesian Inference using Gibbs Sampling. The first two methods Random Sampling and Linear Optimization are biased in the sense that they favor parsimonious solutions, i.e., they prefer to assign high loss rates to a small number of links. The third approach, Bayesian Inference using Gibbs Sampling is based on solid theoretical foundations. The gist is to determine the posterior distribution $\mathbb{P}(\mathbf{X}|\mathbf{Y})$, where $\mathbf{X}$ represents the logarithm of the link loss rates, and $\mathbf{Y}$ represents the logarithm of the end-to-end loss rates. If the distribution is known, we can obtain samples from this distribution, where a sample represents an assignment of loss rates to links that explains the end-to-end measurements. For each link, we compare all sampled loss rates against a threshold, if the link has high loss rates in most samples, then we infer that the link is congested. Since it is very hard to compute the distribution $\mathbb{P}(\mathbf{X}|\mathbf{Y})$ directly, the authors construct a Markov chain whose stationary distribution equals exactly this distribution. If the Markov chain runs sufficiently long, it converges to its stationary distribution, and the samples can be drawn from this distribution. The main problem with this approach is that it is very computationally expensive, and it takes long to obtain the stationary distribution of the underlying Markov chain. For example, the authors could not simulate this approach for a real topology with a realistic number of nodes.

In [Duf06], the author abstracts the properties required in order to be able to apply Boolean network tomography in practice. Boolean network tomography partitions links and paths into good or congested, depending whether their loss rate is below, or respectively, above a given threshold. The threshold is different for links and for paths, and the relation between the two thresholds is of great importance. The key property that enables us to detect a congested link from end-to-end measurements is *separability*, i.e., the Separability Assumption discussed in Section 2.3. A partition is called separable when a path is congested if and only if at least one of its links is congested. Yet in order to find the link-congestion threshold and the path-congestion threshold for a separable partition, the domain for link loss rates cannot be continuous, i.e., there must be a gap between the maximum allowed loss rate of a good link and the minimum allowed loss rate of a congested link. Since *separability* cannot be verified in practice, the author proposes the notion of *weak separability*, for which it is always possible to determine the link-congestion threshold and the path-congestion threshold. A partition is called weakly separable when a path being congested implies that at least one of the links it traverses is congested. Weak

separability means that paths with all good links are correctly identified, but some congested links may go undetected. Finally, the author proposes a simple algorithm, the Smallest Consistent Failure Set (SCFS) inference algorithm, that determines the congested links in a tree topology. The algorithm iteratively infers as congested the link that can explain the largest number of congested paths in the tree. The SCFS algorithm is extended to general topologies in [DTDD07].

COBALT [BMT05] is a heuristic-based inference algorithm that assigns to each link a confidence interval which represents the likelihood of that link to be congested. The confidence interval depends on the number of congested paths the link belongs to, and on the relative loss rates of these paths. The authors compare COBALT with the methods described in [PQW03] and [Duf06], and conclude that when the fraction of congested links increases, the detection rate of congested links drops dramatically for all three approaches.

Most Boolean tomography approaches work under the assumption that all links are equally likely to be congested. However, the work in [NT07a] showed that this assumption is not needed, provided we can take multiple snapshots of the network in order to learn the probability that each link is congested. The authors prove that the probability that each link is congested is statistically identifiable from end-to-end measurements, for all links in the network, if and only if all columns in the routing matrix are distinct. They propose an algorithm which takes as input multiple consecutive snapshots of the network, and computes the probability that each link is congested. The learnt probabilities are then used as prior information, together with the most recent snapshot, to find the congested links using a Maximum A-Posteriori(MAP) estimator. The authors reduce the problem to a convex optimization problem, and propose a heuristic to solve it. Since this approach outputs the most probable solution, it gives inaccurate results when the most probable solution does not include the actual congested links in the network.

2.4.3 Unicast-based Continuous Loss Tomography Methods

The work of [NT07a] brings a new twist in the world of network tomography: one can use multiple consecutive snapshots to gather additional information about the network. This idea gave rise to the question of whether one can use the additional information available from multiple consecutive snapshots to determine the loss rates of links. The first approach that tried to answer this question is the work by [NT07b]. The gist is that losses due to congestion occur

in bursts, and therefore, loss rates of congested links have high variances. Unlike the link loss rates themselves, the variances of the link loss rates are statistically identifiable under certain well-defined conditions. The authors propose an algorithm that takes as input multiple consecutive snapshots, and outputs the variance of the loss rate of each link, for all links in the network. They argue that there exists a monotonic dependence between the mean and the variance of the loss rates of a link, i.e., a link with a very low loss rate variance has a very low loss rate. They suggest a technique that eliminates the links with low variance from the system in Equation 2.2 (it approximates by zero the loss rate of links with low variance) until a system of full column rank is obtained. We can then solve the reduced system to obtain the loss rate of links with high variances. The problem with this approach is that in order to obtain a system of full column rank, we must usually eliminate many links, i.e., approximate their loss rate by zero. This increases the probability that the loss rate of a congested link is approximated by zero. Thus, we end up introducing noise in our measurements which leads to inaccurate results.

Recently, given the advances in compressed sensing, and the similitudes between compress sensing and network tomography, i.e., compress sensing tries to recover a sparse signal, while most work in network tomography assumes congestion is sparse, there is interest to apply compress sensing approaches to the problem of network tomography. The work of [SQZ06] advertises L_1 norm minimization with non-negativity constraints for solving the system in Equation 2.2, under the assumption that most links in the network are good. Though elegant, this technique often mis-classifies a good link as congested, achieving thus a high false positive rate. Nevertheless, combined with other tomographic methods, this technique can significantly increase their accuracy.

In [ZCB06], the authors argue that current tomographic techniques are biased because of the statistical assumptions they make about the network. As opposed to previous work that required various assumptions in order to identify the link characteristics, they propose to determine the characteristics of minimal identifiable sequences of links (MILS), i.e., sequences of links whose properties can be uniquely identified from end-to-end measurements. Thus, this technique cannot compute the loss rate of each individual link, and unfortunately, when dealing with sparse topologies, it is often the case that a MILS represents an entire path. For a finer granularity, this technique can be used as a complement to other approaches, like Bayesian Inference using Gibbs sampling method [PQW03], improving their accuracy.

2.4.4 Non-Tomographic Methods

Other approaches that determine the loss rates or the congestion statuses of links have been proposed in the literature. These methods differ in that they do not attempt to solve the network tomography problem, either continuous loss tomography (Section 2.2) or Boolean loss tomography (Section 2.3). These methods can be divided in two categories: shared congestion techniques and router-based techniques.

Methods that detect shared congestion of flows use the correlations between different flows to identify the shared bottlenecks. The correlations are computed based either on individual probe delivery or on the variations in the throughput of the flows. The cross-correlation between two flows that have a common endpoint is compared against the autocorrelation of each flow. If the former is greater than the latter, then the common links traversed by the two flows are congested. The authors of [RKT02] propose such a technique that detects the points of congestion in the network. This technique is later extended in [HBB00], where new packet probing techniques are considered. More recently, a new method [AdVE07] analyzes throughput correlations among TCP flows in order to infer shared congestion.

Router-based techniques [MSWA03, ZC07] compute the loss rate of links by relying on router support rather than end-to-end measurements. The gist of these methods is that they send special crafted probes directly to the routers located at the endpoints of the link of interest. From the replies of the routers, these techniques compute the loss rate of that particular link. Since these techniques require that each link is measured separately, they do not scale very well to large networks. Furthermore, router-based techniques depend heavily on the cooperation of routers. Unfortunately, for security and performance reasons, many routers do not respond to special crafted probes or limit their response rates to such probes.

2.5 Common Assumptions in Loss Tomography

In this section, we discuss the most common assumptions encountered in network loss tomography. As mentioned in Section 2.1, all tomographic algorithms make two assumptions about the network topology, namely, that all paths remain stable throughout the experiment (the Routing Stability assumption), and that no two links are traversed by the exact same paths (the Link Identifiability assumption). In addition to these two assumptions, multiple-snapshot algo-

rithms require that the behavior of each link can be modeled as a stationary random process (the Stationarity assumption).

Continuous loss tomography and Boolean loss tomography enforce their own specific assumptions. If the goal is to determine the loss rates of links (Section 2.2), then all paths traversing a common link must experience similar performance degradation on that link (the Loss Uniformity assumption) and the loss rates of links are independent (the Link Independence assumption). If the goal is to determine only the congestion statuses of links (see Section 2.3), then the threshold separating good paths from congested ones must be such that a good path implies that all its links are good (the Separability assumption).

Apart from the above general assumptions required to formulate the network tomography problem, each tomographic algorithm makes its own specific assumptions. We discuss below some of the most popular assumptions required by tomographic algorithms.

Assumption 7. ***Probe Correlation:*** *The network supports measurements that require perfect or strong temporal correlation between probes.*

This assumption is demanded by multicast-based approaches or by approaches that emulate multicast traffic with stripes of unicast probes. The strong temporal correlation between probes that belong to different flows ensures similar performance degradation on the links shared by all flows. In the case of a multicast tree, this can overcome the challenge that the loss rates of links are not statistically identifiable from end-to-end measurements.

Assumption 8. ***Sparse Congestion:*** *The percentage of congested links in the network is low.*

The low percentage of congested links in the network plays an important role in tackling the undetermined problem of network tomography. Nevertheless, tomographic algorithms should not assume extremely sparse congestion, otherwise they will produce inaccurate results at a time when they are most needed, i.e., when the network is faced with serious congestion problems.

Assumption 9. ***Link Homogeneity:*** *All links are equally likely to be congested.*

In a heterogeneous network like the Internet, the assumption that all links have the same prior probability of being congested is unrealistic. The work in [ZCB06] reported that links at the core of the network have typically lower probabilities of being congested than links located at the edge of the network. Algorithms

which assume link homogeneity must carefully consider its implications on the accuracy of the results.

Table 2.1 shows a comparison of state-of-the-art tomographic algorithms with respect to the assumptions we have discussed so far. Note that some of these algorithms make other specific assumptions not shown in this table. It is important to note that the assumptions vary in strength, and that depending on the network, some assumptions may hold while others may not be satisfied. Consequently, the algorithm which requires the smallest number of assumptions is not necessary the best for all networks. For example, in Table 2.1, the multicast-based algorithms make the fewest assumptions, however it is extremely hard to enforce in practice the Probe Correlation assumption. Furthermore, some of the algorithms in Table 2.1, i.e., the multicast-based algorithms [CDHT99, CN00, DPPT01] and the SCFS algorithm [Duf06], only work on tree topologies or on forest of trees, but not on mesh topologies. Last but not least, the algorithms differ in the type of information they provide. Continuous tomography algorithms like the multicast-based [CDHT99, CN00, DPPT01], NetDiagnoser [DTDD07], Netquest [SQZ06], and LIA [NT07b] determine the loss rate of links, while the Boolean tomography algorithms like the SCFS [Duf06], MCMC [PQW03], and CLINK [NT07a] determine only if the loss rate of links exceeds a given threshold, and not their actual value.

2.6 Adding the Salt

In the context of network loss tomography, the loss characteristics of links are identifiable if it is theoretically possible to learn their true values from an infinite number of end-to-end measurements. Mathematically, different assignments of the loss characteristics to links must generate different probability distributions of the paths' loss characteristics. Therefore, identifiability is a crucial property of link characteristics in order for inference to be possible. In general, without support for multicast traffic, neither the loss rates nor the congestion status of links are identifiable from end-to-end measurements. Tomographic techniques try to counterbalance this with various assumptions as discussed in Section 2.5.

There exist many sophisticated tomographic algorithms in the literature; they differ in the assumptions they make and the information they provide. These algorithms would be of great use for network debugging, improved quality of service, and automatic recovery from failures. Yet, none of these algorithms

	Single-snapshot				Multiple-snapshot	
	Multicast [emulated] [CDHT99, CN00, DPPT01]	MCMC [PQW03]	SCFS [Duf06], NetDiagnoser [DTDD07]	Netquest [SQZ06]	CLINK [NT07a]	LIA [NT07b]
continuous	×			×		×
Boolean		×	×		×	
Routing Stability	×	×	×	×	×	×
Link Identifiability	×	×	×	×	×	×
Stationarity					×	×
Loss Uniformity		×		×		×
Link Independence	×	×	×	×	×	×
Separability			×		×	
Probe Correlation	×					
Sparse Congestion		×	×	×		×
Link Homogeneity		×	×	×		

Table 2.1: Summary of state-of-the-art tomographic algorithms and their assumptions.

are ever used in practice, because they cannot assess the accuracy of the results they provide. If the assumptions that an algorithm makes are not fulfilled for a given network, then the results may be arbitrarily erroneous. And for most common assumptions in network tomography, there is no practical way to verify if they hold for a given network. Therefore, we argue for tomographic algorithms that work with weaker assumptions, verifiable in practice. We believe this is the fundamental property that can make network tomography practical.

2.7 Conclusions

In this section, we have described the network loss tomography problem, i.e., the inference of link-loss characteristics from end-to-end measurements. We have considered both the continuous and the Boolean versions of the loss tomography problem. We have discussed state-of-the-art tomographic algorithms and compared them with respect to their output and their assumptions. Finally, we have presented our own perspective on network tomography and stated the problems we want to address in our work.

Chapter 3

Netscope: A Link-Loss Inference Algorithm

In this chapter, we focus on continuous loss tomography, whose goal is to determine the loss rates of network links. As we have seen in Section 2.2, this problem is ill-posed, i.e., the loss rates of links are not statistically identifiable from end-to-end measurements.

We propose Netscope, a link-loss inference algorithm, that significantly outperforms the alternatives by using a novel combination of first- and second-order moments of end-to-end measurements. Inspired by previous work [NT07b], we design an algorithm that gains initial information about the network by computing the variances of the loss rates of links and by using these variances as an indication of the congestion level of links, i.e., the more congested the link, the higher the variance of its loss rate. Its novelty lies in the way it uses this information–to identify and characterize the maximum set of links whose loss rates can be accurately inferred from end-to-end measurements.

The rest of this chapter is organized as follows: We discuss the assumptions made by our algorithm in Section 3.1. We give a high-level description of our algorithm in Sections 3.2 and we dive into the details in Section 3.3. We present an accuracy analysis in Section 3.4, evaluate Netscope's performance in Section 3.5, describe a PlanetLab tomographer that runs our algorithm in Section 3.6 and conclude in Section 3.7.

3.1 The Taming of The Assumptions

In this section, we discuss the assumptions required by our link-loss inference algorithm. These assumptions are inherited from the algorithm in [NT07b], which inspired our algorithm.

The Probe Correlation assumption demands multicast-like measurement probes in order to ensure that flows experience similar performance degradation on the shared links. Probe Correlation is a strong assumption since most networks do not support multicast traffic. Fortunately, recent work [NT07b, PQW03, SQZ06] has shown that Probe Correlation is not a necessary assumption in continuous loss tomography. The measurement probes can be unicast, provided that the fraction of probes lost at each link is the same for all paths traversing that link, i.e., the Loss Uniformity assumption, which has been justified for a large enough number of measurement probes [NT07b], holds.

The Link Homogeneity assumption implies that all links, regardless of their nature or of their location in the network, are equally likely to be congested. This assumption may fail in a heterogeneous network like the Internet. For example, links at the edge of the network are more likely to be congested than links located at the core of the network [ZCB06]. Fortunately, previous work has observed that the variance of the loss rate of a link can be used as a good indication of the level of congestion of that link, i.e., the more congested the link, the higher the variance of its loss rate [NT07b].

Unlike the loss rates themselves, the variances of the link loss rates are statistically identifiable from end-to-end measurements if there are no *fluttering paths* in the network [V.P96, NT07b].

Assumption 10. ***No Fluttering Paths:*** *Two paths never meet at one link, diverge, and then meet again at another link.*

Therefore, if two paths share two links e_j and e_k, then they necessarily share all links in between e_j and e_k. We call any two paths that violate this assumption fluttering paths. Fluttering paths may occur as a consequence of load balancing at routers or after a routing failure (during the convergence period to establish a new route). To handle fluterring paths, we must measure the topology frequently, and if such paths appear, we should keep only the measurements on one of the paths.

In order to compute the variances of link loss rates, we must consider multiple consecutive snapshots of the network. As explained in Section 2.1.2, when doing

so, we assume that the behavior of each link can be modeled as a stationary random process, i.e., we make the Stationarity assumption.

Similar to [NT07b], we assume monotonicity of the variances of the link loss rates:

Assumption 11. ***Monotonicity of Link-Loss Variance:*** *For any link e_j, the variance of X_{e_j} is a non-decreasing function of the corresponding link loss rate $1 - \widehat{\phi}_{e_j}$.*

This assumption has been shown to hold based on Internet measurements [V.P96, ZDPS01], including recent PlanetLab experiments with more than two million samples of path loss rates [NT07b].

The Sparse Congestion assumption implies that only few links are responsible for most congested paths in the network. As opposed to previous work that relies heavily on this assumption, we identify m, the maximum number of links for which we can compute the loss rates, and we compute the loss rates of the m most congested links. However, when there are more than m congested links in the network, our inference may become inaccurate. In this sense, we present an accuracy analysis in Section 3.4. Furthermore, we experimentally show in Section 3.5 that our algorithm significantly outperforms the alternatives as the fraction of congested links in the network increases from 5% to 25%.

In conclusion, apart from the basic assumptions discussed in Section 2, which are required in order to formulate continuous loss tomography when using multiple consecutive snapshots of the network, our algorithm assumes No Route Fluttering, Monotonicity of Link-Loss Variance, and Sparse Congestion.

3.2 The Skyline

Our algorithm, Netscope, solves continuous loss tomography, i.e., it takes as input the routing matrix $\mathbf{R}$ and the logarithm of the transmission rates of paths $\mathbf{Y}$, and it outputs the logarithm of the transmission rates of all network links $\mathbf{X}$. More precisely, similar to all other link-loss inference algorithms, it must find the real solution of Equation 2.2. The amount of information available from this equation depends on the properties of the routing matrix: If $\mathbf{R}$ was full-column-rank, we can solve Equation 2.2 and obtain $\mathbf{X}$. If $\mathbf{R}$ is rank-deficient, then there are many different $\mathbf{X}$'s that satisfy Equation 2.2.

Unfortunately, routing matrices are always rank deficient [NGK+09], which means that we need additional information to identify the real solution $\mathbf{X}$. Re-

searchers have proposed three approaches for gathering this additional information: One approach is to assume strong temporal correlation between the measurement probes, achievable in a multicast environment [CDHT99] or with back-to-back probes that emulate multicast [CN00, DPPT01]. Another approach is to formulate continuous loss tomography as an optimization problem: of all possible solutions, pick the one that meets a certain practical constraint, for example, includes the least number of congested links [SQZ06]. A third approach is to first compute the *variances* of link loss rates, then use this information to identify links with negligible loss rate, thereby reducing the number of possible solutions [NT07b]. All these methods are discussed in detail in Section 2.4.

Our intent is to be practical: we want to build an actual "Internet tomographer", i.e., a system that runs on an overlay of Internet hosts, and infers the loss rates of links between them. This leads us away from the (real or emulated) multicast approach. However, Netscope combines elements from both the second and third approaches outlined above with a new technique, in a way that achieves significantly higher accuracy.

The main idea of our algorithm is the following: suppose that we have a network with n links, and an ordering of all these links according to their loss rates. We consider the first k links in this ordering, i.e., the k links least likely to be congested, and we approximate their loss rate by zero. By doing so, we reduce the number of unknowns in Equation 2.2 by k. If k is large enough, Equation 2.2 becomes solvable, and we can compute the loss rate of the remaining $n - k$ links by solving this equation. On the other hand, if k is too large, we might approximate the loss rate of some congested link by zero, and our inference becomes inaccurate (which is exactly the flaw of the algorithm proposed in [NT07b]). Our algorithm identifies the minimum possible k and the *optimal* set of k links such that Equation 2.2 is solvable.

3.3 Under the Microscope

A key concept of our algorithm is the notion of *basis*. We define a basis as a set of links $\mathcal{B} \subseteq E$ whose corresponding columns in the routing matrix $\mathbf{R}$ are linearly independent and which contains exactly $|\mathcal{B}| = Rank(\mathbf{R})$ links. Given a basis $\mathcal{B}$, the set of links E is partitioned into two parts: $\mathcal{B}$ and $\mathcal{K} = E \setminus \mathcal{B}$. If we

know the loss rates of all links in $\mathcal{K}$, we can plug these into Equation 2.2, and then, compute the loss rates of all links in $\mathcal{B}$ by solving:

$$\mathbf{R}^{\mathcal{B}}\mathbf{X}^{\mathcal{B}} = \mathbf{Y} - \mathbf{R}^{\mathcal{K}}\mathbf{X}^{\mathcal{K}} \tag{3.1}$$

where $\mathbf{R}^{\mathcal{K}}$ is the matrix formed by the columns of $\mathbf{R}$ which correspond to the links in $\mathcal{K}$, $\mathbf{X}^{\mathcal{K}}$ is the vector of the logarithm of the transmission rates of links in $\mathcal{K}$, $\mathbf{R}^{\mathcal{B}}$ is the matrix formed by the columns of $\mathbf{R}$ which correspond to the links in $\mathcal{B}$, and $\mathbf{X}^{\mathcal{B}}$ is the vector of the logarithm of the transmission rates of links in $\mathcal{B}$. Unlike Equation 2.2, Equation 3.1 has a unique solution since the fact that $\mathcal{B}$ is a basis implies that $\mathbf{R}^{\mathcal{B}}$ is full-column-rank. Thus, the maximum number of links whose loss rates can be computed by solving Equation 3.1 is $|\mathcal{B}| = Rank(\mathbf{R})$.

For a given network, there exist multiple bases $\mathcal{B} \subseteq E$ and we can solve Equation 3.1 for any of them. The gist is to find a basis $\mathcal{B}$ such that we know the loss rates of all links in $\mathcal{K} = E \setminus \mathcal{B}$. The intuition is the following: Suppose we are told which are the congested links. If we can find a basis $\mathcal{B}$ that contains all congested links, then we can approximate by zero the loss rate of all links in $\mathcal{K}$ and compute the loss rates of all links in $\mathcal{B}$ by solving Equation 3.1. If such a basis does not exist, we can consider the basis $\mathcal{B}$ which contains the most congested links; if the remaining congested links outside $\mathcal{B}$ are few and not highly congested, we can compute the loss rates of the link in $\mathcal{B}$ by solving Equation 3.1. This is precisely what Netscope does, with the difference that it cannot know exactly which are the congested links, thus, it uses an approximate ordering of the links by loss rate, as provided by [NT07b].

Our algorithm consists of three steps. First, it computes an approximate ordering of all links in E by loss rate, as proposed in [NT07b]. Second, it determines a basis $\mathcal{B}$ which includes the most congested links in E as they appear in the ordering from the previous step. Third, it approximates the loss rate of all links in $\mathcal{K} = E \setminus \mathcal{B}$ by zero, and it computes the loss rates of all links in $\mathcal{B}$ by solving Equation 3.1. We now describe each of these steps in more detail. The pesudo-code for Netscope is given by Algorithm 3.1.

3.3.1 Ordering Links by Loss Rate

In this step, we order all links in E by loss rate. The Monotonicity of Link-Loss Variance assumption implies that the less congested a link, the lower the variance of its loss rate. Therefore, an ordering of links by the variance of their

loss rate is also an ordering of links by their loss rate. We have seen in Section 3.1 that, under certain well-defined conditions, the variances of the link loss rates are identifiable. We use the algorithm described in [NT07b] to compute the variance of the loss rate of each link. Next, we order all links in E by decreasing loss rate variance, and we obtain the ordering $\mathcal{O}_E = \langle e_1, e_2, ..., e_{|E|} \rangle$, where e_i has higher variance than e_j, when $j > i$. If the Monotonicity of Link-Loss Variance Assumption holds, then this is also an approximate ordering of links by decreasing loss rate.

3.3.2 Finding the Optimal Basis

The goal of this step is to find the basis $\mathcal{B}$ that contains the most congested links. More precisely, we want to partition E into a basis $\mathcal{B}$ and another set $\mathcal{K}$, such that $\mathcal{B}$ meets the following constraint: given an ordering $\mathcal{O}_E$ of the links by decreasing loss rates, the basis $\mathcal{B}$ maximizes the number of links in $\mathcal{B} \cap \widehat{\mathcal{O}}$, for any prefix $\widehat{\mathcal{O}} =< e_1, e_2, .., e_m >$ of this ordering, where $m \leq |E|$. We call $\mathcal{B}$ the *optimal basis* in the sense that $\mathcal{B}$ contains the most congested links.

In order to compute the optimal basis $\mathcal{B}$, we order the columns of the routing matrix $\mathbf{R}$ according to the ordering $\mathcal{O}_E$ obtained in the previous step, and we apply Gaussian elimination [GL96] to $\mathbf{R}$. The Gaussian elimination algorithm computes a basis for the routing matrix $\mathbf{R}$ by iterating over its columns: if the current column forms a linearly independent set with the columns corresponding to the links already selected for the basis, then we add the corresponding link to the basis; otherwise, we discard this link. The output of the Gaussian elimination algorithm is the optimal basis $\mathcal{B}$ we are searching for. We assign all links which are not in the optimal basis $\mathcal{B}$ to $\mathcal{K}$. Since we iterate starting from the most congested links, our algorithm construct a basis that, from the practical point of view, contains the most congested links.

3.3.3 Computing the Loss Rates of Links

In this step, we compute the loss rates of all links in the optimal basis $\mathcal{B}$, which contains the most congested links. Previously, we have partitioned the set of links E into the optimal basis $\mathcal{B}$ and the set of remaining links $\mathcal{K}$. If the columns in $\mathbf{R}$ correspoding to all congested links are linearly independent, $\mathcal{K}$ contains the least congested links. We approximate the loss rates of all links in $\mathcal{K}$ by zero, and use these values in Equation 3.1. Since matrix $\mathbf{R}^{\mathcal{B}}$ is full-column-rank, we can directly compute the loss rates of all links in $\mathcal{B}$ by solving Equation 3.1 using

Algorithm 3.1 The Netscope Algorithm

Input: E the links in the network
$\mathbf{R}$ the routing matrix
$\mathbf{Y}$ the logarithm of the transmission rates of all paths

Output: $\mathbf{X}$ the logarithm of the transmission rates of all links

Ordering Links by Loss Rate
1 *compute the variance of the loss rate of each link in* E
2 *determine* $\mathcal{O}_E$*, an ordering of links in* E *by decreasing loss rate variance*

Finding the Optimal Basis
3 *arrange the columns in* $\boldsymbol{R}$ *according to* $\mathcal{O}_E$
4 *compute the optimal basis* $\mathcal{B}$ *by applying Gaussian Elimination to* $\boldsymbol{R}$

Computing the Loss Rates
5 *approximate the loss rate of all links in* $\mathcal{K} = E \setminus \mathcal{B}$ *by zero, i.e.,* $\mathbf{X}^{\mathcal{K}} = \mathbf{0}$
6 *compute* $\mathbf{X}^{\mathcal{B}}$ *by minimizing* $\|\mathbf{Y} - \boldsymbol{R}^{\mathcal{B}}\mathbf{X}^{\mathcal{B}}\| + \lambda\|\mathbf{X}^{\mathcal{B}}\|$

return $\mathbf{X} = [\mathbf{X}^{\mathcal{B}}, \mathbf{X}^{\mathcal{K}}]$

standard techniques. However, when approximating the links in $\mathcal{K}$ by zero, we introduced some noise in our measurements. Therefore, we apply "L1 norm minimization with non-negativity constraints" [SQZ06], i.e., we want to minimize the expression $\|\mathbf{Y} - \mathbf{R}^{\mathcal{B}}\mathbf{X}^{\mathcal{B}}\| + \lambda\|\mathbf{X}^{\mathcal{B}}\|$, where λ is a configurable parameter, under the constraint that $\mathbf{X}^{\mathcal{B}}$ has non-negative elements. This optimization chooses an $\mathbf{X}^{\mathcal{B}}$ that may not exactly satisfy Equation. 3.1, but minimizes the corresponding error (hence the first term of the objective function) and favors solutions that involve fewer congested links (hence the second term). We use the default value $\lambda = 0.01$ proposed in [SQZ06], but in our case, this parameter does not play an essential role in computing the loss rates of links, hence, it can also be set to zero.

3.4 Accuracy Analysis

The main source of inaccuracy in our algorithm is the approximation by zero of the loss rate of links outside the optimal basis. We now study the impact of this approximation. In this section only, we abuse the language by saying that links are linearly independent when their corresponding columns in the routing matrix $\mathbf{R}$ are linearly independent.

The performance of our algorithm depends on whether all congested links are linearly independent. If all congested links form a linearly independent set, then the optimal basis found by Netscope contains all of them, and our algorithm correctly sets to zero only the loss rate of good links. However, if some of the congested links are linear combinations of other congested links, then there exists no basis that contains all congested links, and, inevitably, Netscope incorrectly sets to zero the loss rate of some of the congested links. More precisely, it approximates by zero the loss rate of the congested links that (i) are linear combinations of other congested links and (ii) are the least congested. In conclusion, Netscope computes accurately the loss rates of links if all congested links form a linearly independent set. As the number of linearly dependent congested links increases, the performance of our algorithm decreases.

Because the congested links that form a linearly dependent set are the main cause of inaccuracy in our algorithm, we estimate the number of these links and how this number scales with network size. First, we consider the worst-case scenario, that is, the *maximum* number of congested links that are linearly dependent. We know that the *maximum* number of linearly dependent congested links is upper-bounded by the total number of linearly dependent links in a network, which is equal to $|E| - Rank(\mathbf{R})$ for any network. To understand how this number changes with the network size, we studied the scalability properties of the rank of the routing matrix $Rank(\mathbf{R})$, and formally derived a lower bound for it (the proof can be found in [NGK+09]):

$$Rank(\mathbf{R}) \geq |E| - \alpha |V^R| \tag{3.2}$$

where $|V^R|$ is the number of routers in the network and α depends on the network topology, more precisely, on how paths meet and split at each router. This bound tells us that there can be no more than α linearly dependent links per router; these are the links that, if congested, introduce error in our inference. For network topologies collected from the PlanetLab testbed by running traceroute between 400 nodes, α was around 1.2. Our PlanetLab topologies are sparse, because we only consider complete paths, i.e., all the routers on the path must answer to traceroute probes, and we enforce a maximum number of paths per host in order to avoid overloading the network. In practice, topologies may be significantly denser, which implies a smaller value of α. Therefore, there can be no more than a couple of linearly dependent links per router.

Nevertheless, it is unlikely that *all* these "problematic" links happen to be simultaneously congested. Therefore, we also consider the average case scenario,

that is, the *expected* number of congested links which form a linearly dependent set. This quantity depends on the congestion patterns in the network, that is, the localization of failures. We investigate the expected number of linearly dependent congested links for two congestion patterns: (i) "random," where all links have the same probability of being congested, and (ii) "edge," where links located closer to the end-hosts are more likely to be congested. The latter was inspired by the fact that congestion in the Internet happens typically at the edge of the network. For our PlanetLab topologies, we choose a certain fraction of the links in the network which are located either (i) at random locations or (ii) toward the edge of the network, and we compute the expected number of linearly dependent links among the chosen links.

Table 3.1 shows the results for a representative 4000-link topology: if we choose a subset of 400 links from this topology, on average 1.2 of these links are linearly dependent on the others, both when all links have the same probability of being chosen, and when links located closer to the end-hosts are favored. In other words, if 10% of the links are congested, then only 0.3% of these congested links are linearly dependent, hence introduce error in our inference. Moreover, even when the number of congested links increases to 25% of the network links, fewer than 3% of these links are linearly dependent.

	5%	10%	15%	20%	25%
random	0.072	0.326	0.764	1.502	2.53
edge	0.042	0.308	0.696	1.522	2.75

Table 3.1: *Percentage of the expected number of linearly dependent links within a set of chosen links. The links are chosen either at random — "random", or links located closer to the end-hosts are preferred — "edge". The number of chosen links varies from* 5% *to* 25% *of all network links. The results are averaged over* 1000 *runs on a PlanetLab topology with* 4000 *links.*

3.5 Evaluation

Simulator. We have built a simulator, in which the network is represented as a graph, with vertices corresponding to nodes and edges corresponding to links. Each experiment consists of multiple snapshots; unless otherwise stated, we consider 30 snapshots per experiment. In the beginning of each experiment, we chose which links will be congested throughout the experiment by using one of the following methods: (i) "random", where all links are equally likely to be congested, or (ii) "edge", where links located toward the edge of the network are more likely to be congested. The latter setting allows us to simulate scenarios

where congestion happens mostly close to the end-hosts, which is often the case in Internet. In each snapshot, we assign loss rates to all links. We use the same loss model as [NT07b] (also similar to the models used in [PQW03, SQZ06]), which assigns loss rates uniformly distributed between 0 and 0.002 to good links, and between 0.05 and 1 to congested links. We have also experimented with other loss models, but the results were similar. In order to determine the loss rates of paths in each snapshot, we send 1000 packet probes on each path. We determine the packets that get lost at a link through independent Bernoulli processes, i.e., packets are dropped with a fixed probability such that we respect the loss rate assigned to that link. We consider a link whose loss rate is above 0.01 as *lossy* and a link whose loss rate is below this threshold as good.

On the positive side, our simulator captures the fact that the actual loss rates of paths are, in practice, different from the *measured* loss rates of paths. We capture this, because we measure the loss rate of each path as the fraction of packets successfully received along that path, which is what a real measurement tool does. On the negative side, we determine which packets get lost through independent Bernoulli processes, which means that we miss the potential inter-dependencies between successive probes. To our defense, this is the standard way to evaluate tomographic techniques, and simulations with Gilbert losses report similar results.

Topologies. We use real Internet topologies, collected in the following manner: we got hold of as many PlanetLab nodes as we could (400 was the maximum) and ran traceroute between them to identify the set of routers on each path; we discarded all paths with incomplete traceroute results. Even though we repeated the above process several times to collect different topologies, they all look similar in terms of in- and out-degree of the nodes and, hence, yield similar results. Thus, we show results that correspond to one topology of 4000 links (the largest we were able to collect) and note that these are consistent with the results we got from all topologies.

Alternative Techniques. We compare Netscope to other three link-loss inference techniques: "Norm" (L1-norm minimization with non-negativity constraints) [SQZ06], "MultiNorm," a modified version of Norm described below, and "LIA" [NT07b].

Norm is essentially Netscope's third step without the other two (see Section 3.3.1): it simply takes Equation 2.2 and solves it using the norm-minimization approach outlined in Section 3.3.3. Norm applies the "L1-norm minimization with non-negativity constraints" to all the links in the network, while Netscope

applies the minimization only to the links in the optimal basis. A direct comparison between Norm and Netscope would be in some sense unfair, because, unlike Netscope, Norm does not use information from previous snapshots; as a result, Norm more often mis-classifies a good link as lossy. To make a fair comparison, we introduce "MultiNorm," a modified version of Norm, which (just like Netscope and LIA) uses information from previous snapshots: instead of applying L1-norm minimization on *all* the links in the network, it applies it only on links that were lossy in more than $T\%$ of the previous snapshots. Essentially, MultiNorm tries to enforce a certain amount of "stability" across different rounds, i.e., if L1-norm minimization happens to mis-classify a good link as lossy, MultiNorm corrects the mistake, as long as it is infrequent. As expected, MultiNorm's performance depends on the threshold T. We found that $T = 75\%$ gave good results in all our simulation scenarios, hence we show results obtained with this threshold.

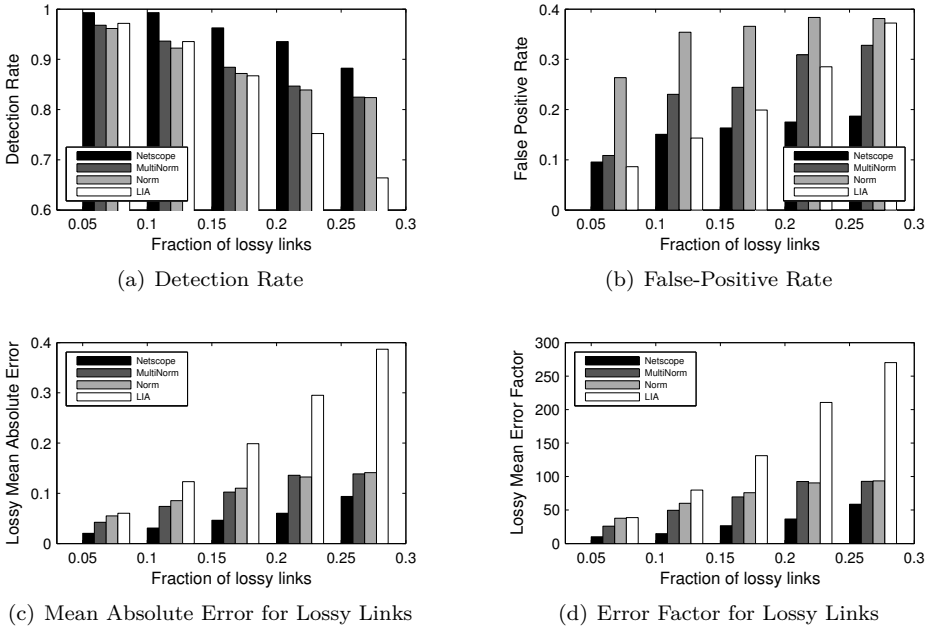
LIA shares the same first step described in Section 3.3.1 with Netscope. In the second step, it partitions the set of links E into two sets, $\mathcal{B}^*$ and $\mathcal{K}^*$, as follows: it goes over the links in E, starting from the lowest-ranked link, i.e., the least congested link according to the ordering obtained in the first step, and greedily removes links until the columns in the routing matrix of the remaining links form a linearly independent set, or a pre-configured minimum number of links is reached; at that point, all the remaining links are assigned to $\mathcal{B}^*$, while all the removed links are assigned to $\mathcal{K}^*$. The problem with the approach taken by LIA is that when it removes links from E, it also implicitly removes some paths, namely, the paths that do not traverse any of the links in $\mathcal{B}^*$. Removing paths from $\mathbf{R}$ is risky as it causes more linear dependence among the columns of the links in $\mathcal{B}^*$. Note that this is different from Netscope's second step: Netscope removes links *optimally*, so as to identify a *basis for* E, without removing any paths. In contrast, LIA removes links *greedily*, until it identifies *any* set of links whose columns in the routing matrix are linearly independent; in many cases, it never identifies such a set and has to stop at an arbitrary point, when the next link to be removed has a loss rate variance above a certain threshold. Therefore, LIA is sensitive to the threshold that specifies when a link has a loss rate variance which is too high in order to approximate its loss rate by zero. As a result, the set $\mathcal{B}^*$ ends up being significantly smaller than $Rank(\mathbf{R})$—which means that LIA freezes significantly more links than necessary. Moreover, LIA uses the least squares method in order to infer the loss rate of links in $\mathcal{B}^*$, which is less accurate than "L1-norm minimization with non-negativity constraints" since the measurements might be noisy.

Netscope is essentially a combination of Norm and LIA plus our optimal-basis selection algorithm described in Section 3.3.2. However, Netscope is robust in the sense that it requires no parameter tuning, unlike MultiNorm which uses a threshold to reduce the number of good links misclassified as lossy, or LIA, which needs a threshold to determine when a link has a high loss rate variance and its loss rate cannot be approximated by zero. Hence, comparing Netscope's performance to the one achieved by each of these two techniques alone is essential in quantifying the value of our contribution.

Metrics. We use four metrics: The *detection rate* specifies the fraction of the lossy links that were correctly identified as lossy. The *false positive rate* specifies the fraction of the links identified as lossy that were actually good. For a given link, the *absolute error* is the absolute difference between the link's actual loss rate and the one inferred by the algorithm. For example, an absolute error of 0.1 means that a link has $L\%$ loss and we incorrectly inferred it has $L \pm 10\%$ loss. For a given link, the *error factor* is the actual loss rate of the link divided by its inferred loss rate, or the other way around, such that the outcome is always larger than 1 [BDPT02]. In our graphs, we show the mean absolute error (and error factor) *of the lossy links only*; otherwise, given that the number of lossy links is relatively small (5% to 25%) the errors in inferring the loss rates of the lossy links would be diluted.

Random Congestion. We look at the performance of the four techniques outlined above for different fractions of lossy links. Fig. 3.1 shows the results when the lossy links are randomly selected. We make the following observations:

When there are only few lossy links, all techniques perform well, with Netscope having a small advantage by all metrics; as the number of lossy links increases, the gap between Netscope's performance and that of the other techniques widens, especially regarding the false-positive rate, the absolute errors, and the error factors. For instance, when 20% of the links are lossy, Netscope detects 94% of the lossy links (MultiNorm, the best alternative, detects 85%) with a false-positive rate of 16% (31% for MultiNorm) and a mean absolute error of 7.5% for the lossy links (14% for MultiNorm). Our interpretation is the following: Of all the $\mathbf{X}$'s that satisfy Equation 2.2, Norm chooses one that minimizes the error between the measurements and the link loss rates, and has the fewest congested links. The larger the number of congested links in the network, the larger the number of different $\mathbf{X}$'s that satisfy Equation 2.2, hence, the less likely it is for Norm to pick the right solution without any additional information. MultiNorm achieves better performance than Norm by using information from previous rounds, however, it still suffers from the same flaw as Norm. In contrast, Netscope uses the



(a) Detection Rate

(b) False-Positive Rate

(c) Mean Absolute Error for Lossy Links

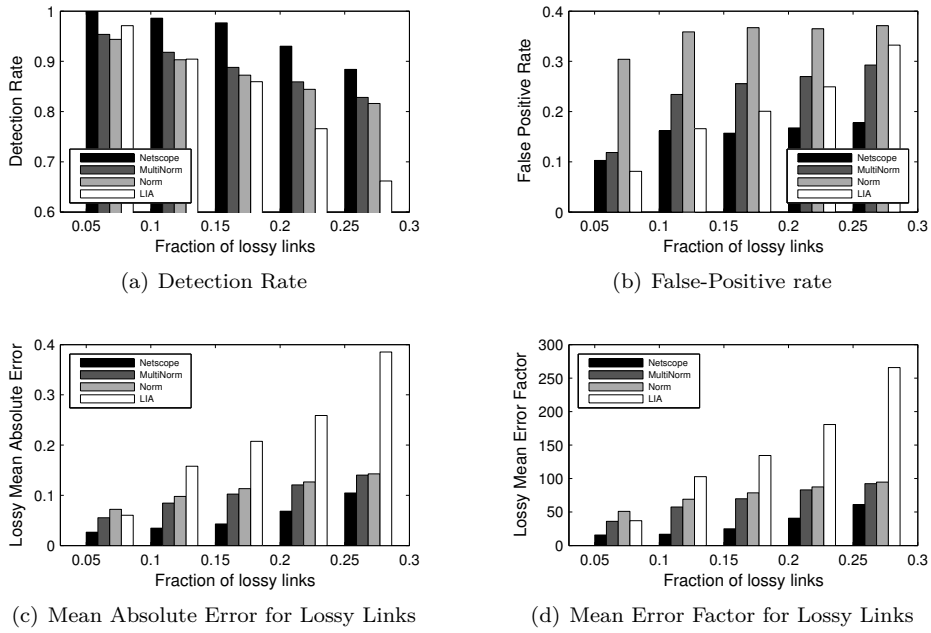
(d) Error Factor for Lossy Links

Figure 3.1: Performance as a function of the fraction of lossy links, when lossy links are randomly chosen.

additional information derived from the variance-based ordering of the links to discard unlikely solutions.

LIA performs well when there are only few lossy links in the network, but its performance degrades quickly as the number of lossy links increases. Among all four techniques, LIA does the worst in terms of identifying the actual link loss rates. Because of its greedy nature, LIA ends up approximating by zero the loss rates of significantly more links than necessary; as the number of lossy links increases, this has a worse impact on performance, because the links whose loss rates is approximated by zero are more likely to be lossy. These results show that it is not enough to just use the additional information provided by the variance-based ordering, it is also important to use it the right way.

Congestion at the Edge. Next, we look at the performance of the four techniques when the lossy links are located closer to the end-hosts. Fig. 3.2 shows the performance as the fraction of lossy links in the network increases. We observe that the gap between Netscope and Norm widens when the lossy links are mostly at the network edge—when 10% of the links are lossy, Norm has more than twice Netscope's false-positive rate, three times its mean absolute error, and five times its mean error factor. This may be due to the fact that



(a) Detection Rate

(b) False-Positive rate

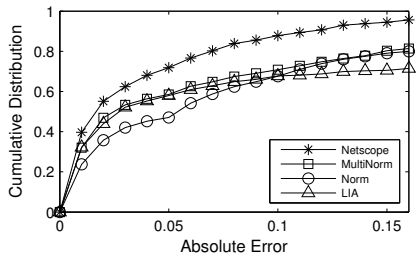
(c) Mean Absolute Error for Lossy Links

(d) Mean Error Factor for Lossy Links

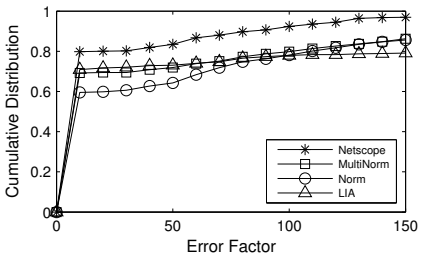
Figure 3.2: Performance as a function of the fraction of lossy links, when lossy links are located closer to the end-hosts.

Norm favors solutions that involve fewer congested links; such solutions tend to involve links that participate in many paths—hence, are not at the edge of the network. MultiNorm performs better than Norm, but its mean absolute error is still two times higher than the one achieved by Netscope.

Fig. 3.3 shows the cumulative distribution function of the absolute errors (the difference between the actual and the inferred loss rates) and the error factors for the lossy links, for the particular case where 15% of the links in the network are lossy. With Netscope, 80% of the lossy links have an absolute error of less than 0.06, which means that, if a link has L% loss, we infer that it has a loss in the range $L \pm 6$%, while the best alternative, MultiNorm, infers that it has a loss in the range $L \pm 16$%, so it is 10% worse in identifying the actual link loss rates. Similarly, for 80% of the lossy links Netscope has an error factor of less than 10, while MultiNorm, the best alternative, achieves an error factor of less than 150.



(a) Cumulative Distribution of the Absolute Error for the Lossy Links



(b) Cumulative Distribution of the Error Factors for the Lossy Links

Figure 3.3: Cumulative distribution function of the absolute error (the difference between the actual and the inferred loss rate) and of the error factors for the lossy links, when 15% of the network links are lossy.

3.6 An Internet Tomographer

We now present an "Internet tomographer", i.e., a distributed system that runs on top of an overlay of PlanetLab nodes [Pla], collects Internet topologies, and infers the loss rates of their links using our algorithm described in Section 3.3. It consists of multiple "beacons" (PlanetLab nodes) that exchange probes and a "manager" (a server) that orchestrates the beacons and collects/processes all their measurements.

The Manager. The manager is the heart of our tomographer: It runs a TCP server that listens for beacon registrations. In the beginning of every snapshot, it contacts all registered beacons, informs them what measurements to perform, and waits for the results. Once all beacons have reported their measurements, the manager runs our link-loss inference algorithm (Algorithm 3.1).

The Beacons. Each beacon runs a TCP and a UDP server, the former for communicating with the manager and the latter for receiving probes from other beacons. When joining the system, the beacon "registers" by sending its IP address and TCP server port number to the manager. In reponse, it receives, in the beginning of each snapshot, the names of a set of target beacons that it must probe. It sends two kinds of probes to each target: (i) traceroute probes, to discover the sequence of links between itself and the target; (ii) UDP probes, to measure the loss rate of the path between them. At the end of the snapshot, the beacon sends the results to the manager; after the first snapshot, to minimize traffic, it only sends incremental updates of the traceroute results.

Traceroute Probing. Each beacon uses traceroute to identify the sequence of links between itself and each target beacon; the manager combines the results

to derive the link-level topology of the network covered by the paths between the communicating beacons. The derived topology may contain inaccuracies, for two reasons: (i) Some Internet routers do not respond to ICMP probes or limit the rate at which they do; as a result we discard all incomplete paths (ii) Routers sometimes respond to different traceroute probes using different IP addresses, depending on the interface on which the probe was received; we use the sr-ally tool to map such addresses to a single router [SWA03], however, that tool does not guarantee 100% accuracy. Due to this factor, what appear as distinct nodes or links in the derived topology may actually correspond to a single node or link.

UDP Probing. Each beacon sends UDP probes to multiple target beacons, a probe carrying a 12-byte sequence number. In order to avoid overloading PlanetLab nodes, a beacon probes around 30 targets, each target once every 10msec. The target beacon that receives UDP probes from a certain source beacon, uses the corresponding sequence numbers to estimate the loss rate of the path between itself and the source; it computes one loss-rate estimate out of 30sec worth of data. We also randomize the order in which beacons probe their targets, to avoid generating traffic bursts on the targets. At the end of each snapshot, all beacons send their path-loss estimates to the manager.

Experiments. The results presented in this section correspond to measurements collected from January 19, 2009 9:40:06 PM to January 26, 2009 11:28:07 AM, which corresponds to 800 measurement snapshots. We engaged about 400 PlanetLab nodes, which resulted in a derived topology of about $9,000$ links and half as many routers. We should clarify that each node acting as a beacon probed about 30 targets, i.e., we did not establish paths between all pairs of nodes. We should also note that many of the participating nodes were highly loaded during the experiments, which led to non-negligible packet loss on (at least) the first and last link of each path; this explains why we observed congested links and paths with more than one congested link.

3.6.1 Validation of Link-Loss Inference

One of the limitations of testing our algorithm on PlanetLab is that we cannot directly measure its accuracy—since we have no way of knowing the actual loss rates that we are estimating. Instead, we use the indirect validation method proposed in [PQW03], where the path loss rates measured in each snapshot are divided into two sets: the *inference set* is used as the input to our link-loss inference algorithm, while the *validation set* is used to validate its results. The

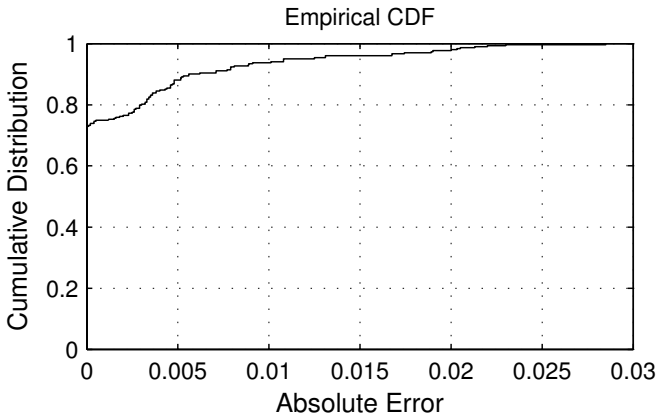


Figure 3.4: Cumulative distribution of the validation error (the difference between the computed and the measured path loss rate) for the PlanetLab experiments.

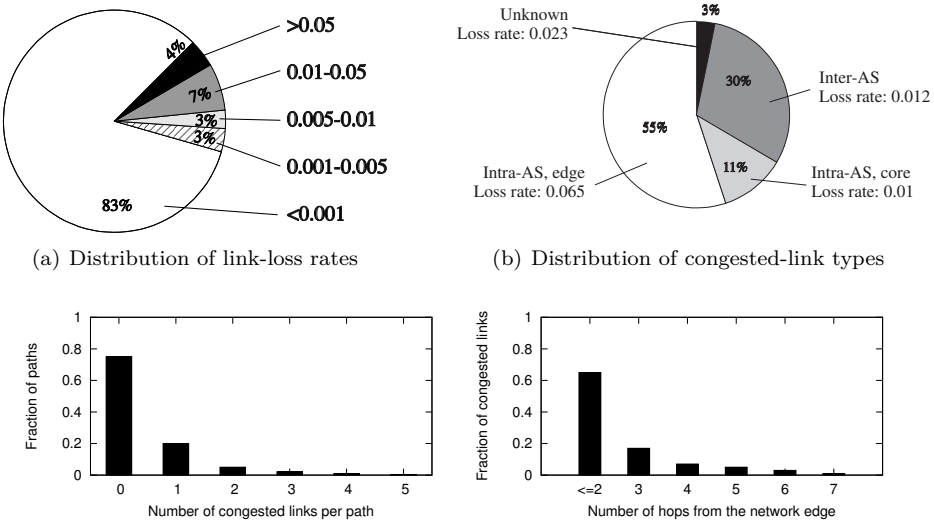
partition is done in such a way that every link is represented in both sets, i.e., for every link $e_j \in E$, there exists at least one path in both sets that includes that link. The link loss rates inferred by our algorithm are then used to compute the path loss rates for the validation paths. For a given validation path, we use the term "validation error" to refer to the difference between the computed (from the inferred link loss rates) and the measured path loss rate.

Figure 3.4 shows the validation error for different paths. We observe that 70% of the paths have 0 validation error, while 94% have a validation error below 1%.

3.6.2 Loss Characteristics of Internet Links

We close with a summary of the statistics we collected using our PlanetLab tomographer: On average, about 83% of links, in each snapshot, had negligible loss rate, while only 4% had a loss rate above 0.05 (Figure 3.5(a)). More than half of the congested links in each snapshot were links located at the network edge; these were also the links with the highest loss rate (Figure 3.5(b)). About 18% of paths had one congested link, while less than 5% had 2 or more congested links (Figure 3.5(c)). Finally, about 62% of congested links were one or two hops away from the network edge, while less than 5% were more than 5 hops away (Figure. 3.5(d)).

We state these without further discussion, as an in-depth analysis of Internet loss characteristics is outside the scope of this thesis. We should also clarify that



(a) Distribution of link-loss rates

(b) Distribution of congested-link types

(c) Distribution of congested links per path

(d) Distribution of distances of congested links from the network edge

Figure 3.5: Loss statistics. The numbers on the y-axes and the pie percentages are averages over 800 snapshots.

these numbers correspond to a period of 7 days and concern a small fraction of Internet links.

3.7 Conclusions

In this chapter, we have presented Netscope, a tomographic algorithm that infers the loss rates of network links from end-to-end measurements. Netscope combines two state-of-the-art approaches with a novel technique in a way that significantly outperforms the existing alternatives. Our algorithm identifies the links with negligible loss rate by computing the variances of the link loss rates, as proposed in [NT07b]. Its novelty lies in the way it uses this information – unlike the algorithm in [NT07b] that greedily infers any link that has a loss rate variance below a certain threshold as non-congested, Netscope identifies an optimal set of links whose loss rates can be inferred from end-to-end measurements. It then applies "L1-norm minimization with non-negativity constraints" as described in [SQZ06] to find the solution that explains the end-to-end measurements. Netscope is robust in the sense that it requires no parameter tunning. Furthermore, the advantage of our algorithm over the existing alternatives increases with the number of congested links in the network.

We have validated Netscope's performance using PlanetLab experiments: We have built a "Internet tomographer" that runs on PlanetLab nodes and infers the loss rates of links located between them; we have used some of the measured paths for inference and others for validation, and we have shown that the results are consistent.

Chapter 4

Boolean Tomography on Correlated Links

In this chapter, we challenge the traditional perspective on network loss tomography. All state-of-the-art tomographic approaches follow a conventional path: Their goal is to infer certain link characteristics that are generally[1] *not statistically identifiable* from end-to-end measurements, and in doing so, they assume that *all links are independent* (the Link Independence assumption described in Section 2.2). In these circumstances, we envision answering the following questions: (Q1) *Is the Link Independence assumption necessary* in order to solve the network tomography problem? and (Q2) *Are there any loss characteristics of links statistically identifiable* from end-to-end measurements if not all links are independent?

The Link Independence assumption is widely used by tomographic algorithms as shown in Table 2.1, but there is no evidence that it holds in practice. In fact, there are practical scenarios in which links are correlated, that is, the losses that occur on one link might depend on the losses that occur on other links in the network. For example, if we know the network topology at the IP-level or at the domain-level, then links in the same local area network, or the same administrative domain, are potentially correlated because they might be sharing physical links, network equipment, and even management processes. When such link correlations are present in the network, the loss characteristics of links inferred by current tomographic algorithms may be inaccurate, moreover, there is no way of knowing to what extent they are inaccurate.

[1] Except in the case of a tree topology, where the loss rates of links are statistically identifiable from end-to-end measurements if the network supports multicast traffic.

We take the first step toward applying network tomography on correlated links. We show that if we partly lift the Link Independence assumption and allow for some link correlations, we can still correctly estimate certain loss characteristics of links. In particular, in the context of Boolean loss tomography, we formally derive the necessary and sufficient condition under which *the probability that each set of links is congested* is statistically identifiable from end-to-end measurements in the presence of link correlations.

Our answers to the raised questions are: (Q1) Under certain well-defined conditions, network tomography works on correlated links; and (Q2) there are certain loss characteristics of links, specifically, the probability that each set of links is congested, that under certain conditions are statistically identifiable from end-to-end measurements even in the presence of correlated links.

The rest of this chapter is organized as follows: We describe several practical scenarios in which links are correlated in Section 4.1. We introduce our link correlation model in Section 4.2 and discuss identifiable link loss characteristics in Section 4.3. We explain the conditions under which the probability that each set of links is congested is identifiable from end-to-end measurements in the presence of correlated links in Section 4.4. We give insights into our theoretical results in Sections 4.5 and 4.6, and finally, prove the theoretical results in Section 4.7.

4.1 The Forgotten Existence of Correlated Links

State-of-the-art tomographic approaches implicitly assume that all links are independent, i.e., the Link Independence assumption discussed in Section 2.2. Nevertheless, there exist practical scenarios in which links are correlated, that is, the losses which occur on one link may depend on the losses which occur on other links in the network. In this section, we describe a few such scenarios.

A factor playing a leading role in the existence of correlated links is incomplete knowledge of the network topology. This affects all tomographic approaches since the loss tomography problem takes as input the network topology.

The ISP curious about its peer. Consider the scenario where the operator of an Internet Service Provider (ISP) wants to estimate the quality of service offered by a peer. Since the operator does not have direct access to the peer's links, she turns to network tomography. The operator measures the loss characteristics on some of the paths which transit the peer, and uses traceroute to discover the underlying network topology, it then applies one of the state-

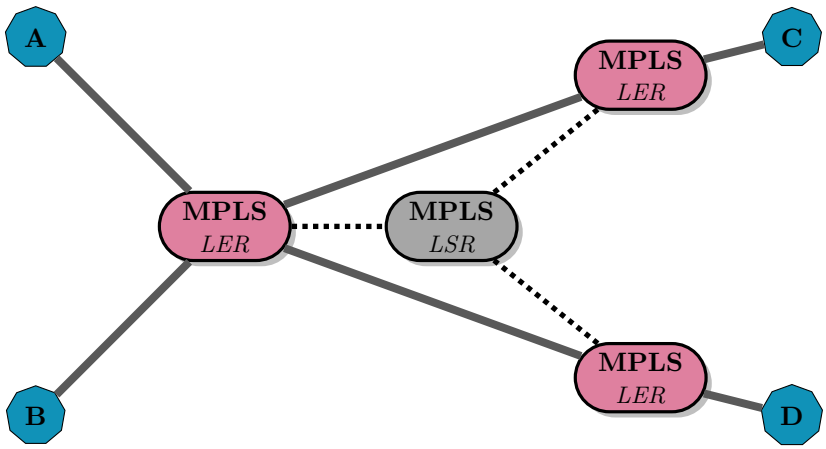


Figure 4.1: An example of correlated links caused by the use of Multiprotocol Label Switching(MPLS) which routes packets based on labels rather than IP addresses. There are two types of MPLS routers: Label Edge Routers (LER) that push/pop labels on packets, and Label Switch Routers (LSR) that perform routing based on these labels. The LSRs remain undiscovered from end-to-end measurements, causing the virtual paths between the LERs to appear as links in the network topology. These links are potentially correlated as they share undiscovered physical links (the dashed lines).

of-the-art tomographic techniques to estimate the loss characteristics of links traversed by the measured paths. But without insider information, she cannot know if all these links are independent. The peer might be using Multiprotocol Label Switching (MPLS) for internal routing, i.e., in order to avoid complex lookups in the routing table, some routers handle packets based on short labels rather than long IP addresses. Each label determines a virtual path, that is, a sequence of physical links, between two distant MPLS-capable routers. There are two types of MPLS routers: Label Edge Routers (LER) that push/pop labels on packets, and Label Switch Routers (LSR) that perform routing based on these labels. As shown in Figure 4.1, the LSRs remain undiscovered from end-to-end measurements; therefore, the virtual paths between the LERs appear as links in the network topology measured by the operator. These links are potentially correlated as they share undiscovered physical links.

Furthermore, it may also be the case that the operator does not *care* to have visibility into the internals of other domains—she is only trying to determine whether the peer is honoring a service-level agreement (SLA). In this case, the network topology is at the granularity of domain-level (as opposed to physical or IP-level): intermediate nodes in the resulting network graph represent border routers, i.e., routers located at the entry/exit points of a domain. Links between

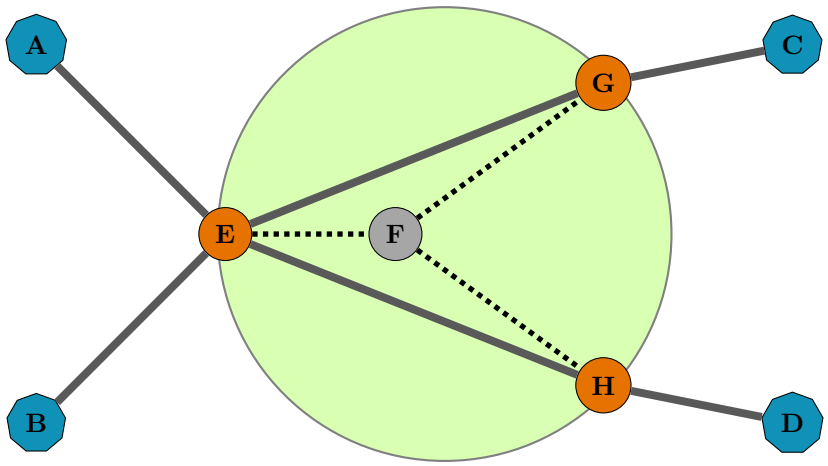


Figure 4.2: An example of correlated links when applying network tomography on a domain-level topology. Inside the domain, only border routers E, G, and H located at the entry/exit points of the domain, and links EG and EH connecting these border routers are visible in the topology. Links EG and EH are correlated as they share the undiscovered physical link between border router E and internal router F.

border routers of the same administrative domain are potentially correlated, because they may be sharing physical links, as well as management processes. For example, in Figure 4.2, inside the domain, only border routers E, G, and H, and links EG and EH connecting these border routers are visible in the topology. Links EG and EH are correlated as they share the undiscovered physical link between border router E and internal router F.

The operator monitoring her network. Even in the scenario of an operator who wants to monitor the quality of links in her own domain in a non-intrusive manner using network tomography, it is still problematic to assume that all links are independent. Suppose the operator relies on traceroute to discover the underlying network topology of her domain. This may seem unreasonable at first as one might assume that an operator already knows the topology of her own domain. Yet, in practice, the operator of a large network, e.g., a university-campus network, does not always have access to all areas and equipment of the network. Moreover, given that paths change in response to network conditions, the operator does not always know which links compose each path. In this scenario, the operator misses all nodes that do not respond to traceroute probes, necessarily including all network elements operating below layer three. In the resulting network graph, nodes represent layer three elements; hence, links between nodes located in the same local-area network

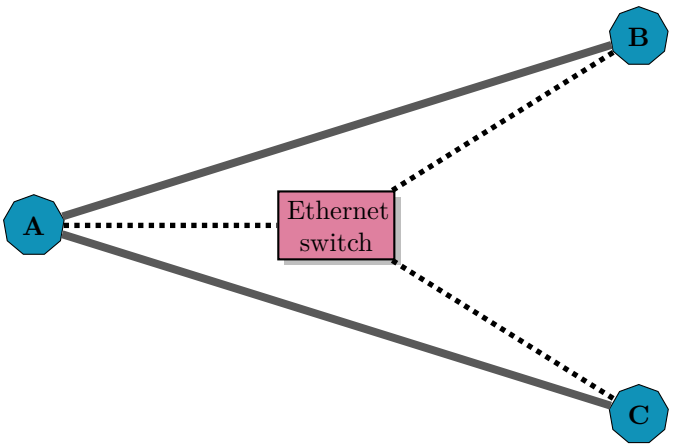


Figure 4.3: An example of correlated links caused by an Ethernet switch in a local-area network. The Ethernet switch operates below layer three; it does not appear in the IP-level topology. Links AB and AC are correlated as they share the physical link between node A and the Ethernet switch.

are potentially correlated, because they may be sharing physical links. This scenario is depicted in Figure 4.3, where links AB and AC located in the same local-area network are correlated as they share the physical link between node A and the Ethernet switch.

In the above scenarios, the network topology is not completely known and may include correlated links. Therefore, the operator using a tomographic technique that assumes Link Independence, cannot assess the accuracy of the estimated loss characteristics of links, it cannot tell if and to what extent these estimates are accurate.

Correlated links are not strictly the consequence of incomplete knowledge of the network topology, they may arise from other causes than MPLS switches or Ethernet switches not visible at layer three. For example, a bad implementation of a network protocol deployed at a router might cause some of its outgoing links to experience synchronous failures, and thus, be correlated; or a denial-of-service attack might cause a particular set of links to be simultaneously congested. To conclude, correlated links do exist in practice; it is unrealistic to discard them and assume that all links are independent.

4.2 Link Correlation Model

The reason why current tomographic algorithms rely on the Link Independence assumption is that network tomography tries to solve a hard problem: both continuous and Boolean loss tomography are ill-posed. Under the perspective that all links are independent, network tomography is more approachable. Given the variety of reasons that cause correlated links, the different nature and degree of link correlations, and the fact that any increase in the number of correlated links implies that we need to consider additional joint probability distributions of the random variables describing the characteristics of these links, it is difficult to study network tomography on correlated links. The challenge is to find a link correlation model that is universal across the various causes of correlated links, and at the same time realistic, but without unnecessarily complicating the network tomography problem. In this section, we propose a model that takes into account correlated links.

Correlated Links. Two links are *statistically independent* (for brevity, just *independent*) if the losses that occur one link are independent from the losses that occur on the other link. In the context of Boolean loss tomography (see Section 2.3), if two links e_j and e_k are independent, then the congestion status of link e_j during a snapshot cannot affect the congestion status of link e_k during the same or any other snapshot; more formally, the random variables Z_{e_j} and Z_{e_k} are independent (or, equivalently, since they are Bernoulli random variables, uncorrelated) from each another. By definition, two links that are not independent, are *correlated*.

Correlation Sets. Unlike previous work which assumes that all links are independent, our work considers a different perspective: we assume that a link may be correlated only with a specific set of other links. More precisely, we partition the set of links E into correlation sets $\{\mathcal{C}_1, \mathcal{C}_2, \dots\}$ such that any two links belonging to different correlation sets are independent. Links within the same correlation set are *potentially correlated*, in the sense that they may be either correlated or independent, but this information is not available to us; hence, to stay on the safe side, we presume that they can be correlated.

Definition 4.0.3. *A correlation set $\mathcal{C}_p$ is a set of potentially correlated links such that:*

$$\forall\, e_j, e_k \in E \text{ s.t. } e_j \in \mathcal{C}_p,\ e_k \notin \mathcal{C}_p,\ Z_{e_j} \text{ is independent from } Z_{e_k}.$$

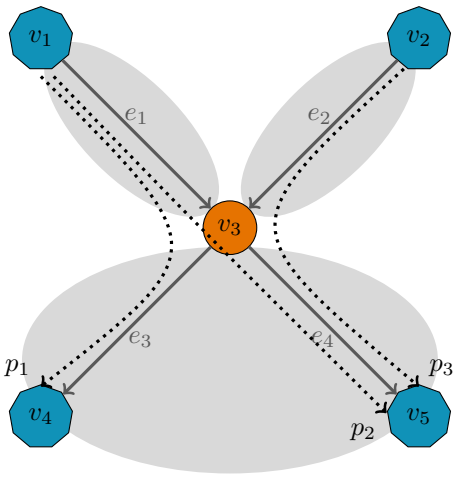


Figure 4.4: A toy topology with correlated links. Hosts $V^H = \{v_1, v_2, v_4, v_5\}$. Routers $V^R = \{v_3\}$. Links $E = \{e_1, e_2, e_3, e_4\}$. Paths $P = \{p_1, p_2, p_3\}$. Correlation sets $C = \{\{e_1\}, \{e_2\}, \{e_3, e_4\}\}$. Correlation subsets $S = \{\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_3, e_4\}\}$.

For example, in Figure 4.4, $\{e_3, e_4\}$ is a correlation set which implies that link e_3 is independent from links e_1 and e_2, but is potentially correlated with link e_4.

Definition 4.0.4. *C is the set of all correlation sets.*

For example, in Figure 4.4, $C = \{\{e_1\}, \{e_2\}, \{e_3, e_4\}\}$. Thus, links e_3 and e_4 are the only potentially correlated links, while links e_1 and e_2 are independent with respect to all other links.

If all links in the network are independent, the set of links E is partitioned into $|E|$ correlation sets, one for each link, i.e., $C = \{\{e_1\}, \{e_2\}, \ldots, \{e_{|E|}\}\}$. At the other extreme, if all links in the network are potentially correlated, there is only one correlation set that includes all links, i.e., $C = \{\{e_1, e_2, \ldots, e_{|E|}\}\}$. Note that it is not a mistake to assign two independent links to the same correlation set since we do not assume that links within the same correlation set are necessarily correlated. On the other hand, our model is violated if two correlated links belong to different correlation sets.

The Correlation Sets Assumption. In our work, the assumption that links are independent is substituted by the assumption that we know the correlation sets.

Assumption 12. ***Correlation Sets:*** *Links are grouped into known correlation sets such that any two links belonging to different correlation sets are independent.*

Our link correlation model is useful in scenarios where the operator knows which links are most likely to be correlated. For instance, consider the scenario where an operator uses network tomography to monitor the quality of links in her domain and relies on traceroute to discover the domain's topology (Scenario "The operator monitoring her network" described in Section 4.1). In this context, it makes sense for the operator to map each local-area network discovered through traceroute to one correlation set. The links in each correlation set are potentially correlated, because they may be sharing physical links and/or management processes.

Alternatively, consider the scenario where the operator of one administrative domain uses network tomography to determine whether a set of neighboring domains are honoring their SLA, but does not have visibility into the internals of these domains, because they use MPLS for internal routing (Scenario "The ISP curious about its peer" described in Section 4.1). In this context, it makes sense for the operator to map each administrative domain to one correlation set. As above, the links in each correlation set are potentially correlated, because they may be sharing physical links and/or management processes.

Nevertheless, our link correlation model is limited in scenarios where the operator does not know which links may be correlated, e.g., when an unpredictable traffic pattern affects the congestion status of multiple otherwise uncorrelated links. For instance, consider an operator that uses network tomography to investigate a denial-of-service attack. Unless the links heavily congested by the attack are part of the same local area network, or the botnet's structure and targets are known, there is no way to guess the link-correlation pattern caused by the attack. Hence, the operator would mislabel these links as uncorrelated, introducing inaccuracies in the model.

Furthermore, in practice, the size of the correlation sets plays an important role. As correlation sets grow in size, it is increasingly more difficult to solve the network tomography problem because we need to consider additional joint probability distributions of the random variables describing the characteristics of links. For instance, under the assumption that all links are potentially correlated, the amount of information provided by network tomography is minimal.

Correlation subset. In our analysis, we often refer to the notion of a *correlation subset*, i.e., a non-empty subset of some correlation set.

Definition 4.0.5. *A correlation subset $\mathcal{S}_k$ is a set of potentially correlated links such that*

$$\mathcal{S}_k \neq \emptyset \text{ and } \mathcal{S}_k \subseteq \mathcal{C}_p \text{ for some } \mathcal{C}_p \in C.$$

Note that a correlation subset belongs to only one correlation set, but a correlation set $\mathcal{C}_p$ has $2^{|\mathcal{C}_p|} - 1$ correlation subsets. For example, in Figure 4.4, the set of links $\{e_3, e_4\}$ is a correlation subset because links e_3 and e_4 belong to the same correlation set. However, the set of links $\{e_1, e_2\}$ is not a correlation subset because links e_1 and e_2 belong to different correlation sets, and are independent. As opposed to a set of independent links where each link may be studied in isolation from the others, a correlation subset may conceal a common cause of congestion.

Definition 4.0.6. *S is the set of all correlation subsets.*

For example, in Figure 4.4, the set of all correlation subsets is $S = \{\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_3, e_4\}\}$.

4.3 Identifiable Link Characteristics

In Chapter 2, we have seen that generally neither the loss rates nor the congestion statuses of links are identifiable from end-to-end measurements, and that current tomographic algorithms try to counterbalance this with various assumptions. We propose a different perspective, namely, we believe that slightly less detailed loss characteristics of links are more useful in practice if they are identifiable from end-to-end measurements.

The work in [NT07a] already gave valuable insights into this topic: in the context of Boolean loss tomography, the probability that a link is congested is identifiable from end-to-end measurements provided that all links are independent, i.e., the Link Independence assumption holds. Nevertheless, as motivated in Section 4.1, we prefer to keep our distance from this assumption since it is a strong premise, violated in practical scenarios. Therefore, we ask the following question: under our correlation model described in Section 4.2, is the probability that each link is congested statistically identifiable from end-to-end measurements? In this section, we sketch the answer to this question with a toy example.

Consider the topology in Figure 4.4. Under the assumption that all links are independent, the tomographic algorithm in [NT07a] forms a system of equations where the unknowns are the probability that each link is congested, for all links in E. Suppose that path p_1 is good, under the Separability assumption, the probability of this event is equal to the probability that both links e_1 and e_3 are good. If the Link Independence assumption holds, this probability is the product of the marginal probabilities forming the first equation in Equation 4.1,

where Z_{e_j} is the congestion status of link e_j given by Definition 2.0.1. In the same way, we can express the probability that each path and each pair of paths is good and form the remaining equations in Equation 4.1.

$$\begin{aligned}
\mathbb{P}(W_{p_1}=0) &= \mathbb{P}(Z_{e_1}=0, Z_{e_3}=0) = \mathbb{P}(Z_{e_1}=0)\,\mathbb{P}(Z_{e_3}=0)\\
\mathbb{P}(W_{p_2}=0) &= \mathbb{P}(Z_{e_1}=0, Z_{e_4}=0) = \mathbb{P}(Z_{e_1}=0)\,\mathbb{P}(Z_{e_4}=0)\\
\mathbb{P}(W_{p_3}=0) &= \mathbb{P}(Z_{e_2}=0, Z_{e_4}=0) = \mathbb{P}(Z_{e_2}=0)\,\mathbb{P}(Z_{e_4}=0)\\
\mathbb{P}(W_{p_2}=0, W_{p_3}=0) &= \mathbb{P}(Z_{e_1}=0, Z_{e_2}=0, Z_{e_4}=0)\\
&= \mathbb{P}(Z_{e_1}=0)\,\mathbb{P}(Z_{e_2}=0)\,\mathbb{P}(Z_{e_4}=0) \qquad (4.1)\\
\mathbb{P}(W_{p_1}=0, W_{p_2}=0) &= \mathbb{P}(Z_{e_1}=0, Z_{e_3}=0, Z_{e_4}=0)\\
&= \mathbb{P}(Z_{e_1}=0)\,\mathbb{P}(Z_{e_3}=0)\,\mathbb{P}(Z_{e_4}=0)\\
\mathbb{P}(W_{p_1}=0, W_{p_3}=0) &= \mathbb{P}(Z_{e_1}=0, Z_{e_2}=0, Z_{e_3}=0, Z_{e_4}=0)\\
&= \mathbb{P}(Z_{e_1}=0)\,\mathbb{P}(Z_{e_2}=0)\,\mathbb{P}(Z_{e_3}=0)\,\mathbb{P}(Z_{e_4}=0)
\end{aligned}$$

The resulting system has four unknowns (one for each link) and four linearly independent equations; hence, we can solve this system using standard linear algebra techniques and determine the probability that each link is good. The probability that a link is congested is the complement with respect to 1 of the probability that the link is good, i.e., $\mathbb{P}\left(Z_{e_j}=1\right) = 1-\mathbb{P}\left(Z_{e_j}=0\right)$, for all links $e_j \in E$. Thus, this algorithm relies on the Link Independence assumption in order to form the system in Equation 4.1 and to compute the probability that each link is congested. If this assumption is violated, the system is incorrect and consequently, the estimated probabilities are inaccurate. For example, in Figure 4.4, if links e_3 and e_4 are indeed correlated, then $\mathbb{P}(Z_{e_3}=0, Z_{e_4}=0) \neq \mathbb{P}(Z_{e_3}=0)\,\mathbb{P}(Z_{e_4}=0)$, and the last two equations of the system in Equation 4.1 are wrong.

However, if we take into account that fact that links e_3 and e_4 are potentially correlated, we obtain the system in Equation 4.2.

As opposed to the system in Equation 4.1 formed under the Link Independence assumption, we have an extra unknown, namely, the probability that links e_3 and e_4 are simultaneously good, i.e., $\mathbb{P}(Z_{e_3}=0, Z_{e_4}=0)$. But if we consider only the first four equations of the system in Equation 4.2, we can ignore the extra unknown, and obtain a system of four linearly independent equations and four unknowns, namely, $\mathbb{P}\left(Z_{e_j}=0\right)$, with $j = 1 \ldots 4$. Thus, even if links e_3 and

e_4 are potentially correlated, we can still determine the probability that each link is congested.

$$\begin{aligned}
\mathbb{P}(W_{p_1}=0) &= \mathbb{P}(Z_{e_1}=0, Z_{e_3}=0) = \mathbb{P}(Z_{e_1}=0)\,\mathbb{P}(Z_{e_3}=0)\\
\mathbb{P}(W_{p_2}=0) &= \mathbb{P}(Z_{e_1}=0, Z_{e_4}=0) = \mathbb{P}(Z_{e_1}=0)\,\mathbb{P}(Z_{e_4}=0)\\
\mathbb{P}(W_{p_3}=0) &= \mathbb{P}(Z_{e_2}=0, Z_{e_4}=0) = \mathbb{P}(Z_{e_2}=0)\,\mathbb{P}(Z_{e_4}=0)\\
\mathbb{P}(W_{p_2}=0, W_{p_3}=0) &= \mathbb{P}(Z_{e_1}=0, Z_{e_2}=0, Z_{e_4}=0)\\
&= \mathbb{P}(Z_{e_1}=0)\,\mathbb{P}(Z_{e_2}=0)\,\mathbb{P}(Z_{e_4}=0) \qquad (4.2)\\
\mathbb{P}(W_{p_1}=0, W_{p_2}=0) &= \mathbb{P}(Z_{e_1}=0, Z_{e_3}=0, Z_{e_4}=0)\\
&= \mathbb{P}(Z_{e_1}=0)\,\mathbb{P}(Z_{e_3}=0, Z_{e_4}=0)\\
\mathbb{P}(W_{p_1}=0, W_{p_3}=0) &= \mathbb{P}(Z_{e_1}=0, Z_{e_2}=0, Z_{e_3}=0, Z_{e_4}=0)\\
&= \mathbb{P}(Z_{e_1}=0)\,\mathbb{P}(Z_{e_2}=0)\,\mathbb{P}(Z_{e_3}=0, Z_{e_4}=0)
\end{aligned}$$

In conclusion, there are scenarios where the probability that each link is congested is identifiable from end-to-end measurements without the need to assume that all links are independent. In the next section, we discuss the identification conditions that make it possible to determine these probabilities.

4.4 Identification Condition

We want to determine whether it is feasible to identify from unicast end-to-end measurements the probability that each link is congested without assuming that all links are independent. Theorem 4.1 provides an answer based on the link correlation model described in Section 4.2, it states that under certain well-defined conditions, the probability that each link is congested is identifiable from end-to-end measurements.

Theorem 4.1. *For any network graph and any partition of links into correlation sets, if Assumptions 1, 3, 6, and 12 (Routing Stability, Stationarity, Separability and Correlation Sets) hold, then the probability that any set of links is congested is identifiable from end-to-end measurements if and only if Condition 1 (Identifiability++) is satisfied.*

Proof. The proof is given in Section 4.7. □

Condition 1. ***Identifiability++:*** *Any two correlation subsets are not traversed by the same paths.*

The Identifiability++ condition generalizes a fundamental condition of network tomography, that any two *links* are not traversed by exactly the same paths, i.e., the Link Identifiability assumption discussed in Section 2.1.1. Intuitively, this earlier condition captured the fact that, when two links participate in exactly the same paths, there is no way to differentiate between the two links based only on end-to-end observations even if all links are independent. We generalize this condition to correlated links to say that, when two *groups of potentially correlated links* participate in exactly the same paths (and assuming we know nothing about the nature of the correlation), there is no way to differentiate between the two groups based only on end-to-end observations. Indeed, in the particular case when all links are independent, our condition becomes exactly the earlier condition.

To better illustrate this condition, we define the path coverage function $Paths(\mathcal{E})$ which maps a set of links $\mathcal{E} \subseteq E$ to the set of paths that traverse at least one of these links.

Definition 4.1.1. *The path coverage function applied to a set of links $\mathcal{E} \subseteq E$ is:*

$$Paths(\mathcal{E}) = \{\ p_i \in P \mid p_i \ni e_j \text{ for some } e_j \in \mathcal{E}\ \}.$$

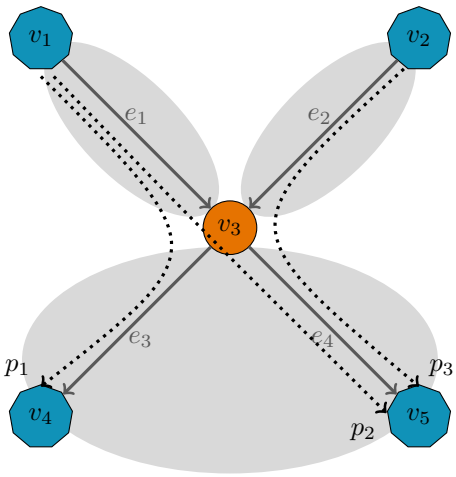
For example, in Figure 4.4, $Paths(\{e_1, e_3\}) = \{p_1, p_2\}$, and $Paths(\{e_1, e_2\}) = \{p_1, p_2, p_3\}$. Using this definition, we can restate the Identifiability++ condition as:

$$\forall\, \mathcal{S}_k, \mathcal{S}_l \in S,\ Paths(\mathcal{S}_k) \neq Paths(\mathcal{S}_l), \tag{4.3}$$

with S given by Definition 4.0.6.

We test whether the Identifiability++ condition holds for the toy topology in Figure 4.5. Indeed, each correlation subset $\mathcal{S}_k \in S$ is traversed by a different set of paths $Paths(\mathcal{S}_k)$. Intuitively, this allows us to measure the probability that the paths which traverse each correlation subset are congested and infer, from that, the probability that the links in each correlation subset are congested.

To illustrate the challenge introduced by link correlations, we also consider the scenario depicted in Figure 4.6, where the Identifiability++ condition does not hold. In this case, correlation subsets $\{e_1\}$ and $\{e_2, e_3\}$ are traversed by the same set of paths $\{p_1, p_2\}$. As a result, we cannot distinguish between the probability that e_1 is congested and the probability that e_2 and e_3 are simulta-



$\mathcal{S}_k \in S$	$Paths(\mathcal{S}_k)$
$\{e_1\}$	$\{p_1, p_2\}$
$\{e_2\}$	$\{p_3\}$
$\{e_3\}$	$\{p_1\}$
$\{e_4\}$	$\{p_2, p_3\}$
$\{e_3, e_4\}$	$\{p_1, p_2, p_3\}$

Figure 4.5: A toy topology with correlated links where the Identifiability++ condition holds, i.e., each correlation subset $\mathcal{S}_k$ is traversed by a different set of paths $Paths(\mathcal{S}_k)$. Hosts $V^H = \{v_1, v_2, v_4, v_5\}$. Routers $V^R = \{v_3\}$. Links $E = \{e_1, e_2, e_3, e_4\}$. Paths $P = \{p_1, p_2, p_3\}$. Correlation sets $\mathcal{C}_1 = \{e_1\}$, $\mathcal{C}_2 = \{e_2\}$, and $\mathcal{C}_3 = \{e_3, e_4\}$. Correlation subsets $S = \{\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_3, e_4\}\}$.

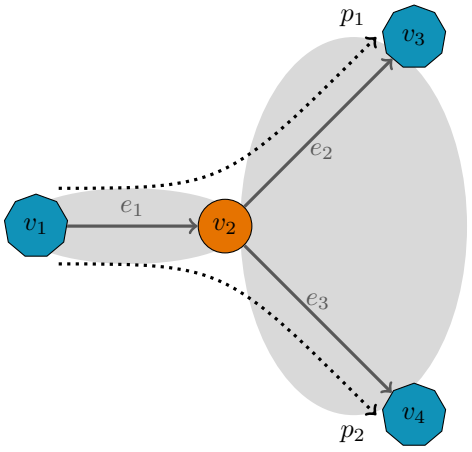
neously congested. If links e_2 and e_3 were uncorrelated, we would not have this problem; we could form the system of equations as explained in Section 4.3:

$$\mathbb{P}(W_{p_1} = 0) = \mathbb{P}(Z_{e_1} = 0)\,\mathbb{P}(Z_{e_2} = 0)$$
$$\mathbb{P}(W_{p_2} = 0) = \mathbb{P}(Z_{e_1} = 0)\,\mathbb{P}(Z_{e_3} = 0)$$
$$\mathbb{P}(W_{p_1} = 0, W_{p_2} = 0) = \mathbb{P}(Z_{e_1} = 0)\,\mathbb{P}(Z_{e_2} = 0)\,\mathbb{P}(Z_{e_3} = 0)$$

which has three unknowns, i.e., $\mathbb{P}\left(Z_{e_j} = 0\right)$ with $j = 1 \ldots 3$, and three linearly independent equations; thus, we can solve this system and determine the probability that each link is congested. Nevertheless, in our example links e_2 and e_3 are correlated, hence, the correct system of equations is:

$$\mathbb{P}(W_{p_1} = 0) = \mathbb{P}(Z_{e_1} = 0)\,\mathbb{P}(Z_{e_2} = 0)$$
$$\mathbb{P}(W_{p_2} = 0) = \mathbb{P}(Z_{e_1} = 0)\,\mathbb{P}(Z_{e_3} = 0)$$
$$\mathbb{P}(W_{p_1} = 0, W_{p_2} = 0) = \mathbb{P}(Z_{e_1} = 0)\,\mathbb{P}(Z_{e_2} = 0, Z_{e_3} = 0)$$

which has four unknowns, i.e., $\mathbb{P}\left(Z_{e_j} = 0\right)$ with $j = 1 \ldots 3$, and $\mathbb{P}(Z_{e_2} = 0, Z_{e_3} = 0)$, but only three linearly independent equations. Therefore, this system is undetermined, and in this case, we cannot compute accurately the probability that each link is congested. This un-identifiability problem is not a consequence of



$\mathcal{S}_k \in S$	$Paths(\mathcal{S}_k)$
$\{e_1\}$	$\{p_1, p_2\}$
$\{e_2\}$	$\{p_1\}$
$\{e_3\}$	$\{p_2\}$
$\{e_2, e_3\}$	$\{p_1, p_2\}$

Figure 4.6: A toy topology with correlated links where the Identifiability++ condition does not holds, i.e., each correlation subset $\mathcal{S}_k$ is not traversed by a different set of paths $Paths(\mathcal{S}_k)$. Hosts $V^H = \{v_1, v_3, v_4\}$. Routers $V^R = \{v_2\}$. Links $E = \{e_1, e_2, e_3\}$. Paths $P = \{p_1, p_2\}$. Correlation sets $\mathcal{C}_1 = \{e_1\}$, and $\mathcal{C}_2 = \{e_2, e_3\}$. Correlation subsets $S = \{\{e_1\}, \{e_2\}, \{e_3\}, \{e_2, e_3\}\}$.

the algorithm we are using, but is intrinsic to the network tomography problem. The Identifiability++ condition is both necessary and sufficient in order to identify the probability that each set of links is congested.

4.5 Congestion Probability

One one hand, the probability that a link is congested is less detailed information than the congestion status or the loss rate of the link. As opposed to the congestion status of a link, which provides the exact times when the link was congested or the loss rate of the link, which represents the fraction of packets lost at that link, the congestion probability of a link tells us only *how often* the link is congested. For example, instead of knowing that link e_j was congested precisely 20 minutes ago (in the case of the congestion status) or that it lost 5% of the packets in the last hour (in the case of the loss rate), we only know that link e_j was congested 15% of the time during the last hour. On the other hand, the probability that each link is congested can be obtained accurately under weaker assumptions than those required by current tomographic algorithms as stated by Theorem 4.1.

The probability that each link is congested represents valuable information for network diagnosis, routing algorithms and SLA verification. Consider the

scenario where the operator of an ISP wants to monitor the quality of service offered by its most important peers. In particular, for each peer, the operator wants to understand: how often the peer is responsible for connectivity/performance problems encountered by the customers; how frequently the peer is congested and how its congestion level changes over the course of a day or a week. If the operator knows the probability that each intra-domain link inside the peer's network is congested, she already has the answer to these questions. In short, this information is very powerful for an ISP since it enables the operator to take the right decisions with respect to its peers.

Theorem 4.1 establishes the identification conditions not only for the probability that each individual link is congested, i.e., $\mathbb{P}\left(Z_{e_j}=1\right)$ for all $e_j \in E$, but also for the probability that all links in a set are simultaneously congested, i.e., $\mathbb{P}\left(\cap_{e_j\in\mathcal{E}}\left\{Z_{e_j}=1\right\}\right)$ for all $\mathcal{E}\subseteq E$. We call the probability that link e_j is congested, *the congestion probability of link* e_j, and the probability that all links in a set $\mathcal{E}\subseteq E$ are congested, *the congestion probability of link set* $\mathcal{E}$. Compared to the congestion probabilities of individual links, the congestion probabilities of link sets provide additional information about the network at no additional measurement cost, because the same end-to-end measurements are used. For example, the congestion probabilities of link sets give valuable insights into which network links are actually correlated, and how strong their correlation is. Remember that the correlation sets defined in Section 4.2 have may-semantics: links in the same correlation set may be correlated, but they do not need to be. If we know which links in the network are actually correlated, we can improve the network diagnosis phase. For instance, if the probability that all outgoing links of a router are simultaneously congested is high, then most likely something is wrong with the router itself. Furthermore, if the Correlation Sets assumption holds, we can use the congestion probabilities of link sets to reduce the size of the correlation sets. Finally, the congestion probabilities of link sets would be useful for routing algorithms, e.g., in order to compute disjoint paths which do not traverse links that tend to fail together.

If the Correlation Sets assumption holds, then links belonging to different correlation sets are independent; hence, we can express the congestion probability of link set $\mathcal{E}$ as:

$$\mathbb{P}\left(\bigcap_{e_j\in\mathcal{E}}\left\{Z_{e_j}=1\right\}\right)=\prod_{\mathcal{C}_p\in C}\mathbb{P}\left(\bigcap_{e_j\in\mathcal{E}\cap\mathcal{C}_p}\left\{Z_{e_j}=1\right\}\right),$$

with C given by Definition 4.0.4. A set of links $\mathcal{E} \cap \mathcal{C}_p$ is either the empty set or a correlation subset since it belongs to correlation set $\mathcal{C}_p$. In the latter case, $(\mathcal{E} \cap \mathcal{C}_p) \in S$ with S given by Definition 4.0.6. For example, in Figure 4.5, we have $\mathbb{P}(Z_{e_1} = 1, Z_{e_2} = 1) = \mathbb{P}(Z_{e_1} = 1)\,\mathbb{P}(Z_{e_2} = 1)$ as links e_1 and e_2 belong to different correlation sets and are independent, but $\mathbb{P}(Z_{e_1} = 1, Z_{e_3} = 1, Z_{e_4} = 1) = \mathbb{P}(Z_{e_1} = 1)\,\mathbb{P}(Z_{e_3} = 1, Z_{e_4} = 1)$ as links e_3 and e_4 are potentially correlated. Therefore, the congestion probability of a link set can be expressed as a product of the congestion probabilities of various correlation subsets.

In conclusion, if we know the congestion probability of correlation subset $\mathcal{S}_k$, for all $\mathcal{S}_k \in S$, then it is straightforward to compute the congestion probability of all possible sets of links. For example, in Figure 4.4, we need to know the congestion probability of each individual link, i.e., $\mathbb{P}\left(Z_{e_j} = 1\right)$ with $j = 1 \ldots 4$, and the probability $\mathbb{P}(Z_{e_3} = 1, Z_{e_4} = 1)$. In the particular case when all links are independent, it is sufficient to know the congestion probability of each link in order to able to compute the congestion probability of all possible sets of links.

4.6 Illustration of Theoretical Results

In this section, we illustrate how the proof of Theorem 4.1 works using the topology in Figure 4.5 where the Identifiability++ condition holds, and the topology in Figure 4.6 where the Identifiability++ condition does not hold. Our goal is to identify the congestion probabilities of all sets of links from the probabilities that sets of paths are congested, where the latter are available from end-to-end measurements. As explained in Section 4.5, the quantities of interest can be computed if we know the congestion probability of each correlation subset $\mathcal{S}_k \in S$, with $\mathcal{S}_k$ given by Definition 4.0.5. Hence, a key contribution of the proof of Theorem 4.1 consists of showing that we can, indeed, compute all these $|S|$ probabilities, if and only if the Identifiability++ condition holds.

4.6.1 Definitions and Notations

We start by defining and providing compact notation for certain terms that appear frequently in our illustration. All defined symbols are summarized in Table 4.1.

Definition 4.1.2. *The network state* $\boldsymbol{S}$ *is the set of all congested links during a snapshot:*

$$\boldsymbol{S} \equiv \{\ e_j \in E \mid Z_{e_j} = 1\ \},$$

with Z_{e_j} *given by Definition 2.0.1.*

For example, in Figure 4.5, if links e_1 and e_3 are good, whereas e_2 and e_4 are congested, then $\mathbf{S} = \{e_2, e_4\}$. Under the Separability assumption, a congested path must traverse at least one congested link. From Definition 4.1.1 of the path coverage function, $Paths(\mathbf{S})$ is equal to the set of all congested paths during a snapshot, i.e.,

$$Paths(\mathbf{S}) = \{\ p_i \in P \mid W_{p_i} = 1\ \}, \tag{4.4}$$

with W_{p_i} given by Definition 2.0.2. For example, in Figure 4.5, if $\mathbf{S} = \{e_2, e_4\}$, then $Paths(\mathbf{S}) = \{p_2, p_3\}$.

Definition 4.1.3. *The state* $\boldsymbol{S}_{\mathcal{C}_p}$ *of correlation set* $\mathcal{C}_p$ *is the set of all congested links in* $\mathcal{C}_p$ *during a snapshot:*

$$\boldsymbol{S}_{\mathcal{C}_p} \equiv \{\ e_j \in \mathcal{C}_p \mid Z_{e_j} = 1\ \},$$

with $\mathcal{C}_p$ *given by Definition 4.0.3, and* Z_{e_j} *given by Definition 2.0.1.*

The network state is the union of the states of all correlation sets, i.e.,

$$\mathbf{S} = \bigcup_{\mathcal{C}_p \in C} \mathbf{S}_{\mathcal{C}_p}. \tag{4.5}$$

with C the set of all correlation sets (Definition 4.0.4). For example, in Figure 4.5, if links e_1 and e_3 are good, whereas e_2 and e_4 are congested, then $\mathbf{S}_{\mathcal{C}_1} = \emptyset$, $\mathbf{S}_{\mathcal{C}_2} = \{e_2\}$, and $\mathbf{S}_{\mathcal{C}_3} = \{e_4\}$. Since for any two links that belong to different correlation sets $e_j \in \mathcal{C}_p$ and $e_k \in \mathcal{C}_q$, $p \neq q$, the congestion statuses Z_{e_j} and Z_{e_k} are statistically independent (Definition 4.0.3), it is also the case that the states of any two correlation sets $\mathbf{S}_{\mathcal{C}_p}$ and $\mathbf{S}_{\mathcal{C}_q}$, $p \neq q$, are statistically independent.

Given a correlation subset $\mathcal{S}_k \in S$ that belongs to a correlation set $\mathcal{C}_p$, i.e., $\mathcal{S}_k \subseteq \mathcal{C}_p$, we refer to the following events:

Definition 4.1.4. $\boldsymbol{S}_{\mathcal{C}_p} = \mathcal{S}_k$ *is the event that the links in correlation subset* $\mathcal{S}_k$ *are the only congested links in correlation set* $\mathcal{C}_p$*.*

For example, in Figure 4.5, $\mathbf{S}_{\mathcal{C}_3} = \{e_4\}$ is the event that link e_4 is congested, whereas link e_3 is good.

Definition 4.1.5. *$Paths(\boldsymbol{S}) = Paths(\mathcal{S}_k)$ is the event that the paths traversing links in correlation subset $\mathcal{S}_k$ are the only congested paths in the network.*

For example, in Figure 4.5, $Paths(\mathbf{S}) = Paths(\{e_4\})$ is the event that paths p_2 and p_3 are congested, whereas p_1 is good.

Finally, for each correlation subset $\mathcal{S}_k \subseteq \mathcal{C}_p$, we define its *congestion factor* $\alpha_{\mathcal{S}_k}$.

Definition 4.1.6. *The congestion factor $\alpha_{\mathcal{S}_k}$ of correlation subset $\mathcal{S}_k$ is:*

$$\alpha_{\mathcal{S}_k} = \frac{\mathbb{P}\left(\boldsymbol{S}_{\mathcal{C}_p} = \mathcal{S}_k\right)}{\mathbb{P}\left(\boldsymbol{S}_{\mathcal{C}_p} = \emptyset\right)}.$$

This expresses how often the links in $\mathcal{S}_k$ are the only congested links in correlation set $\mathcal{C}_p$ compared to how often *all* links in $\mathcal{C}_p$ are good. Note that $\mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \emptyset\right) \neq 0$ because there is a non-zero probability that all paths in the network are good.

4.6.2 The Identifiability++ condition is sufficient.

Using the example in Figure 4.5, we first illustrate that, if the Identifiability++ condition holds, then we can identify the probability that all links in each correlation subset are congested, i.e., $\mathbb{P}\left(Z_{e_j} = 1\right)$ with $j = 1 \ldots 4$, and $\mathbb{P}\left(Z_{e_3} = 1, Z_{e_4} = 1\right)$.

Setup. Consider the event that all paths are good. By the Separability assumption, this implies that all links are good. Using the definitions in Section 4.6.1, we obtain:

$$\mathbb{P}\left(Paths(\mathbf{S}) = \emptyset\right) = \mathbb{P}\left(\mathbf{S} = \emptyset\right) = \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_1} = \emptyset\right) \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_2} = \emptyset\right) \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_3} = \emptyset\right). \quad (4.6)$$

From end-to-end measurements, we can measure the probability that all paths are good.

We consider each correlation subset $\mathcal{S}_k \in S$ and the event that the paths traversing links in $\mathcal{S}_k$ are the only congested paths in the network, that is, the event $Paths(\mathbf{S}) = Paths(\mathcal{S}_k)$. From end-to-end measurements, we can also measure the probabilities of these events.

Step 1. Consider the event that $Paths(\{e_3\}) = \{p_1\}$ is the only congested path in the network. Since paths p_2 and p_3 are good, then links e_1, e_2 and e_4 are good, and link e_3 must be congested, i.e., the network can only be in state $\mathbf{S} = \{e_3\}$:

$\mathbf{S}_{\mathcal{C}_1}$	$\mathbf{S}_{\mathcal{C}_2}$	$\mathbf{S}_{\mathcal{C}_3}$	$\mathbf{S}$	$Paths(\mathbf{S})$
$\emptyset$	$\emptyset$	$\{e_3\}$	$\{e_3\}$	$\{p_1\}$

Hence, we can write:

$$\mathbb{P}\left(Paths(\mathbf{S}) = \{p_1\}\right) = \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_1} = \emptyset\right)\ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_2} = \emptyset\right)\ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_3} = \{e_3\}\right).$$

If we divide this by Equation 4.6, we get

$$\frac{\mathbb{P}\left(\ Paths(\mathbf{S}) = \{p_1\}\ \right)}{\mathbb{P}\left(\ Paths(\mathbf{S}) = \emptyset\ \right)} \ = \ \frac{\mathbb{P}\left(\ \mathbf{S}_{\mathcal{C}_3} = \{e_3\}\ \right)}{\mathbb{P}\left(\ \mathbf{S}_{\mathcal{C}_3} = \emptyset\ \right)} \ = \ \alpha_{\{e_3\}},$$

where $\alpha_{\{e_3\}}$ is the congestion factor of $\{e_3\}$ given by Definition 4.1.6. Since both the numerator and denominator of the left-most term can be measured, we can compute $\alpha_{\{e_3\}}$.

Step 2. Consider the event that $Paths(\{e_1\}) = \{p_1, p_2\}$ are the only congested paths in the network. Since path p_3 is good, links e_2 and e_4 are good; hence, either e_1 is the only congested link, or e_1 and e_3 are the only congested links, i.e., the network can only be in state $\mathbf{S} = \{e_1\}$ or in state $\mathbf{S} = \{e_1, e_3\}$:

$\mathbf{S}_{\mathcal{C}_1}$	$\mathbf{S}_{\mathcal{C}_2}$	$\mathbf{S}_{\mathcal{C}_3}$	$\mathbf{S}$	$Paths(\mathbf{S})$
$\{e_1\}$	$\emptyset$	$\emptyset$	$\{e_1\}$	$\{p_1, p_2\}$
$\{e_1\}$	$\emptyset$	$\{e_3\}$	$\{e_1, e_3\}$	$\{p_1, p_2\}$

Therefore, we can write:

$$\begin{aligned}\mathbb{P}\left(\ Paths(\mathbf{S}) = \{p_1, p_2\}\ \right) = \ & \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_1} = \{e_1\}\right)\ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_2} = \emptyset\right)\ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_3} = \emptyset\right) \\ & + \ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_1} = \{e_1\}\right)\ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_2} = \emptyset\right)\ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_3} = \{e_3\}\right).\end{aligned}$$

If we divide this by Equation 4.6, we get

$$\frac{\mathbb{P}\left(\ Paths(\mathbf{S}) = \{p_1, p_2\}\ \right)}{\mathbb{P}\left(\ Paths(\mathbf{S}) = \emptyset\ \right)} \ = \ (1 + \alpha_{\{e_3\}})\ \alpha_{\{e_1\}}.$$

Since both the numerator and denominator of the left-most term can be measured, and we have already computed $\alpha_{\{e_3\}}$, we can now compute $\alpha_{\{e_1\}}$.

Step 3. With the same rationale, we compute all congestion factors $\alpha_{\mathcal{S}_k}$, for all correlation subsets $\mathcal{S}_k \in S$. The gist is that, thanks to the Identifiability++ condition, we can *order* the correlation subsets and compute their congestion factors such that each factor depends only on terms that can be measured or have already been computed. In our particular example, a possible ordering is $\langle\{e_3\}, \{e_2\}, \{e_1\}, \{e_4\}, \{e_3, e_4\}\rangle$.

Step 4. According to Lemma 4.3, once we have computed all congestion factors, we can derive $\mathbb{P}\left(Z_{e_j} = 1\right)$ for all $j = 1 \ldots 4$, and $\mathbb{P}\left(Z_{e_3} = 1, Z_{e_4} = 1\right)$. Once we know these 5 probabilities, we can easily compute the rest since links which belong to different correlation sets are independent, e.g., $\mathbb{P}\left(Z_{e_1} = 1, Z_{e_3} = 1\right) = \mathbb{P}\left(Z_{e_1} = 1\right)\ \mathbb{P}\left(Z_{e_3} = 1\right)$.

4.6.3 The Identifiability++ condition is necessary.

Using the example in Figure 4.6, we now illustrate that, if the Identifiability++ condition does not hold, then we cannot always identify the probability that each set of links is congested from end-to-end measurements.

Setup. As in the previous example, we can measure the probability that all paths are good, i.e.,

$$\mathbb{P}\left(\mathit{Paths}(\mathbf{S}) = \emptyset\right) \;=\; \mathbb{P}\left(\mathbf{S} = \emptyset\right) \;=\; \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_1} = \emptyset\right)\ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_2} = \emptyset\right). \tag{4.7}$$

Step 1. Consider the event that $\mathit{Paths}(\{e_2\}) = \{p_1\}$ is the only congested path in the network. Exactly as in the previous example, we can divide the probability that p_1 is the only congested path in the network, by Equation 4.7, and compute $\alpha_{\{e_2\}}$. Similarly, we can consider the event that $\mathit{Paths}(\{e_3\}) = \{p_2\}$ is the only congested path in the network, and compute $\alpha_{\{e_3\}}$.

Step 2. Consider the event that $\mathit{Paths}(\{e_1\}) = \{p_1, p_2\}$ are both congested. In this case, the network can be in one of the following states:

$\mathbf{S}_{\mathcal{C}_1}$	$\mathbf{S}_{\mathcal{C}_2}$	$\mathbf{S}$	$\mathit{Paths}(\mathbf{S})$
$\{e_1\}$	$\emptyset$	$\{e_1\}$	$\{p_1, p_2\}$
$\{e_1\}$	$\{e_2\}$	$\{e_1, e_2\}$	$\{p_1, p_2\}$
$\{e_1\}$	$\{e_3\}$	$\{e_1, e_3\}$	$\{p_1, p_2\}$
$\{e_1\}$	$\{e_2, e_3\}$	$\{e_1, e_2, e_3\}$	$\{p_1, p_2\}$
$\emptyset$	$\{e_2, e_3\}$	$\{e_2, e_3\}$	$\{p_1, p_2\}$

Hence, we can write:

$$\begin{aligned}\mathbb{P}\left(\mathit{Paths}(\mathbf{S}) = \{p_1, p_2\}\right) &= \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_1} = \{e_1\}\right)\ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_2} = \emptyset\right) \\ &+ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_1} = \{e_1\}\right)\ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_2} = \{e_2\}\right) \\ &+ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_1} = \{e_1\}\right)\ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_2} = \{e_3\}\right) \\ &+ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_1} = \{e_1\}\right)\ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_2} = \{e_2, e_3\}\right) \\ &+ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_1} = \emptyset\right)\ \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_2} = \{e_2, e_3\}\right).\end{aligned}$$

If we divide this by Equation 4.7, we get:

$$\frac{\mathbb{P}\left(\mathit{Paths}(\mathbf{S}) = \{p_1, p_2\}\right)}{\mathbb{P}\left(\mathit{Paths}(\mathbf{S}) = \emptyset\right)} = \alpha_{\{e_1\}}(1 + \alpha_{\{e_2\}} + \alpha_{\{e_3\}} + \alpha_{\{e_2,e_3\}}) + \alpha_{\{e_2,e_3\}}.$$

This is the only equation where the congestion factors $\alpha_{\{e_1\}}$ and $\alpha_{\{e_2,e_3\}}$ appear, i.e., we have only one equation for two unknowns. Furthermore, there is no additional information that we can obtain from end-to-end measurements: In the context of Boolean loss tomography, we can only measure the congested paths in the network. Hence, we must compute the congestion factors from the probabilities that sets of paths are congested. In the case that $\mathbb{P}\left(\mathit{Paths}(\mathbf{S}) = \{p_1, p_2\}\right) \neq 0$, we cannot compute all congestion factors.

4.7 Theoretical Results

In this section, we introduce additional definitions and notations that we will later use in order to prove Theorem 4.1. All our symbols are summarized in Table 4.1.

4.7.1 A Partial Ordering of Correlation Subsets

Definition 4.1.7. *The precedence relation between two correlation subsets $\mathcal{S}_k, \mathcal{S}_l \in S$ is:*

$$\mathcal{S}_k \prec \mathcal{S}_l \ \equiv\ |\mathit{Paths}(\mathcal{S}_k)| < |\mathit{Paths}(\mathcal{S}_l)|,$$

where $|\mathit{Paths}(\mathcal{S}_k)|$ is the number of paths traversing links in $\mathcal{S}_k$ given by Definition 4.1.1.

The fact that $\mathcal{S}_k \prec \mathcal{S}_l$ implies that the links in $\mathcal{S}_k$ are traversed by fewer paths than the links in $\mathcal{S}_l$. For example, in Figure 4.5, we have $\mathit{Paths}(\{e_3\}) = \{p_1\}$,

and $Paths(\{e_1\}) = \{p_1, p_2\}$; since $|Paths(\{e_3\})| = 1 < 2 = |Paths(\{e_1\})|$, we obtain $\{e_3\} \prec \{e_1\}$.

Definition 4.1.8. *$\mathcal{O}_S$ is a partial ordering of all the correlation subsets in S induced by the precedence relation given by Definition 4.1.7.*

That is, the correlation subsets in $\mathcal{O}_S$ are ordered by the number of paths that traverse links in these subsets. In Figure 4.5, a possible partial ordering is $\mathcal{O}_S = \{\{e_3\}, \{e_2\}, \{e_1\}, \{e_4\}, \{e_3, e_4\}\}$.

4.7.2 Some Basic Probabilities

In this section, we formally express the probabilities of the basic events defined in Section 4.6.1. We will use these probabilities in the proof of Theorem 4.1.

Network State Probability

From Definitions 4.1.2, and 4.1.3,

$$\mathbf{S}_{\mathcal{C}_p} = \mathbf{S} \cap \mathcal{C}_p, \text{ for all } \mathcal{C}_p \in C. \tag{4.8}$$

When $\mathbf{S} = \mathcal{E}$, with $\mathcal{E} \subseteq E$, i.e., only the links in $\mathcal{E}$ are congested, Equation 4.8 yields $\mathbf{S}_{\mathcal{C}_p} = \mathcal{E} \cap \mathcal{C}_p$, for all correlation sets $\mathcal{C}_p \in C$. A set of links $\mathcal{E} \cap \mathcal{C}_p$ is either the empty set when no links in $\mathcal{E}$ belong to correlation set $\mathcal{C}_p$, or a correlation subset since it is included in correlation set $\mathcal{C}_p$. For example, in Figure 4.5, suppose links e_1 and e_3 are good, whereas links e_2 and e_4 are congested, then the network state is $\mathbf{S} = \{e_2, e_4\}$, and the states of the correlation sets are $\mathbf{S}_{\mathcal{C}_1} = \mathbf{S} \cap \{e_1\} = \emptyset$, $\mathbf{S}_{\mathcal{C}_2} = \mathbf{S} \cap \{e_2\} = \{e_2\}$, and $\mathbf{S}_{\mathcal{C}_3} = \mathbf{S} \cap \{e_3, e_4\} = \{e_4\}$.

We consider the probability that the network is in state $\mathbf{S} = \mathcal{E}$, with $\mathcal{E} \subseteq E$. Since we assume independence between correlation sets, i.e., the Correlation Sets assumption, from Equations 4.5 and 4.8, we get:

$$\mathbb{P}(\ \mathbf{S} = \mathcal{E}\ ) = \mathbb{P}\left(\bigcap_{\mathcal{C}_p \in C} \mathbf{S}_{\mathcal{C}_p} = \mathcal{E} \cap \mathcal{C}_p\right) = \prod_{\mathcal{C}_p \in C} \mathbb{P}\left(\ \mathbf{S}_{\mathcal{C}_p} = \mathcal{E} \cap \mathcal{C}_p\ \right). \tag{4.9}$$

Next, we express the ratio of the probability that the network is in state $\mathbf{S} = \mathcal{E}$ to the probability that all links in the network are good, i.e., $\mathbf{S} = \emptyset$. From Equation 4.9, and Definition 4.1.6, we obtain:

$$\frac{\mathbb{P}\left(\mathbf{S} = \mathcal{E}\right)}{\mathbb{P}\left(\mathbf{S} = \emptyset\right)} = \frac{\prod_{\mathcal{C}_p \in C} \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \mathcal{E} \cap \mathcal{C}_p\right)}{\prod_{\mathcal{C}_q \in C} \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_q} = \emptyset\right)} = \prod_{\mathcal{C}_p \in C} \frac{\mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \mathcal{E} \cap \mathcal{C}_p\right)}{\mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \emptyset\right)}$$

$$= \prod_{\substack{\mathcal{C}_p \in C \\ \text{s.t. } \mathcal{E} \cap \mathcal{C}_p \neq \emptyset}} \alpha_{\mathcal{E} \cap \mathcal{C}_p}. \tag{4.10}$$

All Paths Are Good

Consider the event that all paths in P are good, i.e., $Paths(\mathbf{S}) = \emptyset$ (Definition 4.1.5). We express the probability of this event using the Separability assumption, in particular, its implication that, if all paths in P are good, then necessarily all links in E are good. From Equation 4.9, we obtain:

$$\mathbb{P}\left(Paths(\mathbf{S}) = \emptyset\right) = \mathbb{P}\left(\mathbf{S} = \emptyset\right) = \prod_{\mathcal{C}_p \in C} \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \emptyset\right). \tag{4.11}$$

Some Paths Are Congested

Consider the event that all paths traversing links in correlation subset $\mathcal{S}_k$ are congested, while all other paths in the network are good, i.e., $Paths(\mathbf{S}) = Paths(\mathcal{S}_k)$ (Definition 4.1.5). We express the probability of this event as:

$$\mathbb{P}\left(Paths(\mathbf{S}) = Paths(\mathcal{S}_k)\right) = \sum_{\substack{\mathcal{E} \subseteq E \text{ s.t.} \\ Paths(\mathcal{E}) = Paths(\mathcal{S}_k)}} \mathbb{P}\left(\mathbf{S} = \mathcal{E}\right). \tag{4.12}$$

Clearly, one possible set of links is $\mathcal{E} = \mathcal{S}_k$, i.e., the links in $\mathcal{S}_k$ are the only congested links in the network, but there may be other sets of links which are spread across different correlation sets. Therefore, we can rewrite Equation 4.12 as:

$$\mathbb{P}\left(Paths(\mathbf{S}) = Paths(\mathcal{S}_k)\right) = \mathbb{P}\left(\mathbf{S} = \mathcal{S}_k\right) + \sum_{\substack{\mathcal{E} \subseteq E, \mathcal{E} \neq \mathcal{S}_k \text{ s.t.} \\ Paths(\mathcal{E}) = Paths(\mathcal{S}_k)}} \mathbb{P}\left(\mathbf{S} = \mathcal{E}\right). \tag{4.13}$$

We now develop Equation 4.13 further by considering the following: Since $\mathcal{S}_k$ is a correlation subset, there must be one correlation set that contains $\mathcal{S}_k$,

which we denote by $\mathcal{C}_q$, i.e., $\mathcal{S}_k \subseteq \mathcal{C}_q$. We partition all possible network states $\mathcal{E} \subseteq E$, for which $\mathcal{E} \neq \mathcal{S}_k$ and $Paths(\mathcal{E}) = Paths(\mathcal{S}_k)$, into two sets: the states where $\mathcal{E} \cap \mathcal{C}_q = \mathcal{S}_k$ (the links in $\mathcal{S}_k$ are the only congested links in correlation set $\mathcal{C}_q$) and the states where $\mathcal{E} \cap \mathcal{C}_q \neq \mathcal{S}_k$:

$$\begin{aligned}\mathbb{P}\left(Paths(\mathbf{S}) = Paths(\mathcal{S}_k)\right) = \mathbb{P}\left(\ \mathbf{S} = \mathcal{S}_k\ \right) \\ + \sum_{\substack{\mathcal{E}\subseteq E, \mathcal{E}\neq\mathcal{S}_k \text{ s.t. } \mathcal{E}\cap\mathcal{C}_q=\mathcal{S}_k, \\ Paths(\mathcal{E})=Paths(\mathcal{S}_k)}} \mathbb{P}\left(\ \mathbf{S} = \mathcal{E}\ \right) \\ + \sum_{\substack{\mathcal{E}\subseteq E \text{ s.t. } \mathcal{E}\cap\mathcal{C}_q\neq\mathcal{S}_k, \\ Paths(\mathcal{E})=Paths(\mathcal{S}_k)}} \mathbb{P}\left(\ \mathbf{S} = \mathcal{E}\ \right). \end{aligned} \tag{4.14}$$

Note that the condition $\mathcal{E} \cap \mathcal{C}_q \neq \mathcal{S}_k$ ensures that $\mathcal{E} \neq \mathcal{S}_k$ since $\mathcal{S}_k \subseteq \mathcal{C}_q$. Furthermore, if we use Equation 4.9, we obtain:

$$\begin{aligned}&\mathbb{P}\left(\ Paths(\mathbf{S}) = Paths(\mathcal{S}_k)\ \right) = \mathbb{P}\left(\ \mathbf{S}_{\mathcal{C}_q} = \mathcal{S}_k\ \right) \prod_{\mathcal{C}_p\in C, p\neq q} \mathbb{P}\left(\ \mathbf{S}_{\mathcal{C}_p} = \emptyset\ \right) \\ &+\ \mathbb{P}\left(\ \mathbf{S}_{\mathcal{C}_q} = \mathcal{S}_k\ \right) \sum_{\substack{\mathcal{E}\subseteq E, \mathcal{E}\neq\mathcal{S}_k \text{ s.t. } \mathcal{E}\cap\mathcal{C}_q=\mathcal{S}_k, \\ Paths(\mathcal{E})=Paths(\mathcal{S}_k)}} \left(\prod_{\mathcal{C}_p\in C, p\neq q} \mathbb{P}\left(\ \mathbf{S}_{\mathcal{C}_p} = \mathcal{E}\cap\mathcal{C}_p\ \right) \right) \\ &+ \sum_{\substack{\mathcal{E}\subseteq E \text{ s.t. } \mathcal{E}\cap\mathcal{C}_q\neq\mathcal{S}_k, \\ Paths(\mathcal{E})=Paths(\mathcal{S}_k)}} \left(\prod_{\mathcal{C}_p\in C} \mathbb{P}\left(\ \mathbf{S}_{\mathcal{C}_p} = \mathcal{E}\cap\mathcal{C}_p\ \right) \right). \end{aligned} \tag{4.15}$$

Finally, if we divide by Equation 4.11, we obtain:

$$\frac{\mathbb{P}\left(\ Paths(\mathbf{S}) = Paths(\mathcal{S}_k)\ \right)}{\mathbb{P}\left(\ Paths(\mathbf{S}) = \emptyset\ \right)} = \alpha_{\mathcal{S}_k}(1 + \Gamma_{\mathcal{S}_k}) + \Gamma_{\bar{\mathcal{S}}_k} \tag{4.16}$$

where

$$\Gamma_{\mathcal{S}_k} = \sum_{\substack{\mathcal{E}\subseteq E, \mathcal{E}\neq\mathcal{S}_k \text{ s.t. } \mathcal{E}\cap\mathcal{C}_q=\mathcal{S}_k, \\ Paths(\mathcal{E})=Paths(\mathcal{S}_k)}} \left(\prod_{\substack{\mathcal{C}_p\in C, p\neq q \\ \mathcal{E}\cap\mathcal{C}_p\neq\emptyset}} \alpha_{\mathcal{E}\cap\mathcal{C}_p} \right), \tag{4.17}$$

and

$$\Gamma_{\bar{\mathcal{S}}_k} = \sum_{\substack{\mathcal{E}\subseteq E \text{ s.t. } \mathcal{E}\cap\mathcal{C}_q\neq\mathcal{S}_k, \\ Paths(\mathcal{E})=Paths(\mathcal{S}_k)}} \left(\prod_{\substack{\mathcal{C}_p\in C \\ \mathcal{E}\cap\mathcal{C}_p\neq\emptyset}} \alpha_{\mathcal{E}\cap\mathcal{C}_p} \right). \tag{4.18}$$

Illustration

Consider the toy topology in Figure 4.5, correlation subset $\mathcal{S}_k = \{e_3, e_4\}$, and the event ($Paths(\mathbf{S}) = Paths(\mathcal{S}_k) = \{p_1, p_2, p_3\}$), i.e., all paths are congested. In this case, the network is in one of the following states:

$\mathbf{S}$	$\mathbf{S}_{\mathcal{C}_1}$	$\mathbf{S}_{\mathcal{C}_2}$	$\mathbf{S}_{\mathcal{C}_3}$
$\mathcal{E}_1 = \{e_3, e_4\}$	$\emptyset$	$\emptyset$	$\{e_3, e_4\}$
$\mathcal{E}_2 = \{e_1, e_3, e_4\}$	$\{e_1\}$	$\emptyset$	$\{e_3, e_4\}$
$\mathcal{E}_3 = \{e_2, e_3, e_4\}$	$\emptyset$	$\{e_2\}$	$\{e_3, e_4\}$
$\mathcal{E}_4 = \{e_1, e_2, e_3, e_4\}$	$\{e_1\}$	$\{e_2\}$	$\{e_3, e_4\}$
$\mathcal{E}_5 = \{e_1, e_2\}$	$\{e_1\}$	$\{e_2\}$	$\emptyset$
$\mathcal{E}_6 = \{e_1, e_2, e_3\}$	$\{e_1\}$	$\{e_2\}$	$\{e_3\}$
$\mathcal{E}_7 = \{e_1, e_2, e_4\}$	$\{e_1\}$	$\{e_2\}$	$\{e_4\}$
$\mathcal{E}_8 = \{e_1, e_4\}$	$\{e_1\}$	$\emptyset$	$\{e_4\}$

In this particular case, $\mathcal{S}_k \subseteq \mathcal{C}_3$, that is, $q = 3$. The first state is $\mathcal{E}_1 = \mathcal{S}_k = \{e_3, e_4\}$. For the next three states $\mathcal{E}_i$ with $i = 2 \dots 4$, we have $\mathcal{E}_i \cap \mathcal{C}_3 = \mathcal{S}_k$ and $\mathcal{E}_i \neq \mathcal{S}_k$, whereas for the rest of the states $\mathcal{E}_i$ with $i = 5 \dots 8$, we have $\mathcal{E}_i \cap \mathcal{C}_3 \neq \mathcal{S}_k$. Hence, if we apply Equations 4.14 and 4.15, we obtain:

$$\mathbb{P}(\ Paths(\mathbf{S}) = Paths(\{e_3, e_4\})\ ) = \mathbb{P}(\mathbf{S} = \mathcal{E}_1) + \sum_{i=2}^{4} \mathbb{P}(\mathbf{S} = \mathcal{E}_i) + \sum_{i=5}^{8} \mathbb{P}(\mathbf{S} = \mathcal{E}_i)$$

$$\begin{aligned}
= \quad & \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_1} = \emptyset\ ) \quad \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_2} = \emptyset\ ) \quad \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_3} = \{e_3, e_4\}\ ) \\
+ \quad & \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_1} = \{e_1\}\ ) \quad \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_2} = \emptyset\ ) \quad \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_3} = \{e_3, e_4\}\ ) \\
+ \quad & \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_1} = \emptyset\ ) \quad \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_2} = \{e_2\}\ ) \quad \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_3} = \{e_3, e_4\}\ ) \\
+ \quad & \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_1} = \{e_1\}\ ) \quad \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_2} = \{e_2\}\ ) \quad \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_3} = \{e_3, e_4\}\ ) \\
+ \quad & \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_1} = \{e_1\}\ ) \quad \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_2} = \{e_2\}\ ) \quad \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_3} = \emptyset\ ) \\
+ \quad & \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_1} = \{e_1\}\ ) \quad \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_2} = \{e_2\}\ ) \quad \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_3} = \{e_3\}\ ) \\
+ \quad & \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_1} = \{e_1\}\ ) \quad \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_2} = \{e_2\}\ ) \quad \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_3} = \{e_4\}\ ) \\
+ \quad & \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_1} = \{e_1\}\ ) \quad \mathbb{P}\left(\ \mathbf{S}^2 = \emptyset\ \right) \quad \mathbb{P}(\ \mathbf{S}_{\mathcal{C}_3} = \{e_4\}\ ).
\end{aligned}$$

When we divide by Equation 4.11, we obtain:

$$\begin{aligned}\frac{\mathbb{P}(\ \mathit{Paths}(\mathbf{S}) = \mathit{Paths}(\{e_3, e_4\})\ )}{\mathbb{P}(\ \mathit{Paths}(\mathbf{S}) = \emptyset\ )} &= \alpha_{\{e_3,e_4\}}(1 + \Gamma_{\{e_3,e_4\}}) + \Gamma_{\overline{\{e_3,e_4\}}} \\ &= \alpha_{\{e_3,e_4\}}(1 + \underbrace{\alpha_{\{e_1\}} + \alpha_{\{e_2\}} + \alpha_{\{e_1\}}\alpha_{\{e_2\}}}_{\Gamma_{\{e_3,e_4\}}}) \\ &+ \underbrace{\alpha_{\{e_1\}}\alpha_{\{e_2\}}(1 + \alpha_{\{e_3\}} + \alpha_{\{e_4\}}) + \alpha_{\{e_1\}}\alpha_{\{e_4\}}}_{\Gamma_{\overline{\{e_3,e_4\}}}}.\end{aligned}$$

4.7.3 Proof of Theorem 4.1

In this section, we present the proof of Theorem 4.1 which states that under certain well-defined conditions, the congestion probability of link set $\mathcal{E}$ is identifiable from end-to-end measurements, for all possible link sets $\mathcal{E} \subseteq E$. First, we introduce the lemmas used in this proof.

Lemma 4.2. *For any correlation subset $\mathcal{S}_k \in S$, $\Gamma_{\mathcal{S}_k}$ and $\Gamma_{\bar{\mathcal{S}}_k}$ given by Equations 4.17 and 4.18, depend only on congestion factors $\alpha_{\mathcal{S}_l}$ (Definition 4.1.6) of correlation subsets $\mathcal{S}_l \in S$ such that $\mathcal{S}_l \prec \mathcal{S}_k$ (Definition 4.1.7).*

Proof. Consider a correlation subset $\mathcal{S}_k \in S$ (Definition 4.0.5). From Equations 4.17, and 4.18, we know that $\Gamma_{\mathcal{S}_k}$ and $\Gamma_{\bar{\mathcal{S}}_k}$ depend only on congestion factors $\alpha_{\mathcal{E}\cap\mathcal{C}_p}$, where $\mathcal{C}_p$ is a correlation set (Definition 4.0.3), and $\mathcal{E} \subseteq E$ is a set of links such that $\mathcal{E} \neq \mathcal{S}_k$ and $\mathit{Paths}(\mathcal{E}) = \mathit{Paths}(\mathcal{S}_k)$, with $\mathit{Paths}(\mathcal{E})$ the path coverage function given by Definition 4.1.1.

First, we show that all correlation subsets $\mathcal{S}_l \neq \mathcal{S}_k$ whose congestion factors $\alpha_{\mathcal{S}_l}$ appear in $\Gamma_{\mathcal{S}_k}$ or $\Gamma_{\bar{\mathcal{S}}_k}$, satisfy $\mathcal{S}_l \prec \mathcal{S}_k$. In order for $\alpha_{\mathcal{S}_l}$ to appear in $\Gamma_{\mathcal{S}_k}$ or $\Gamma_{\bar{\mathcal{S}}_k}$, $\mathcal{S}_l = \mathcal{E}\cap\mathcal{C}_p$, where $\mathcal{C}_p$ is the correlation set of $\mathcal{S}_l$, and $\mathcal{E} \subseteq E$ satisfies $\mathit{Paths}(\mathcal{E}) = \mathit{Paths}(\mathcal{S}_k)$. Therefore, $\mathit{Paths}(\mathcal{S}_l) = \mathit{Paths}(\mathcal{E} \cap \mathcal{C}_p) \subseteq \mathit{Paths}(\mathcal{E})$, and consequently, $\mathit{Paths}(\mathcal{S}_l) \subseteq \mathit{Paths}(\mathcal{S}_k)$. We distinguish two cases:

(a) $\mathit{Paths}(\mathcal{S}_l) = \mathit{Paths}(\mathcal{S}_k)$. Correlation subsets $\mathcal{S}_l$ and $\mathcal{S}_k$ are traversed by the same paths. But according to Identifiability++ condition, there exist no two correlation subsets traversed by the same paths. Hence, it must be the case that $\mathcal{S}_l = \mathcal{S}_k$, which contradicts our hypothesis.

(b) $\mathit{Paths}(\mathcal{S}_l) \subset \mathit{Paths}(\mathcal{S}_k)$. Correlation subset $\mathcal{S}_l$ is traversed by fewer paths than $\mathcal{S}_k$, i.e., $|\mathit{Paths}(\mathcal{S}_l)| < |\mathit{Paths}(\mathcal{S}_k)|$. Therefore, we obtain $\mathcal{S}_l \prec \mathcal{S}_k$.

Second, we show that the congestion factor $\alpha_{\mathcal{S}_k}$ cannot appear in $\Gamma_{\mathcal{S}_k}$ and $\Gamma_{\bar{\mathcal{S}}_k}$. We denote by $\mathcal{C}_q$ the correlation set of $\mathcal{S}_k$. From Equation 4.17, $\Gamma_{\mathcal{S}_k}$ depends

only on congestion factors $\alpha_{\mathcal{E}\cap\mathcal{C}_p}$, where $p \neq q$, hence, the congestion factors of correlation subsets that belong to $\mathcal{C}_q$ do not appear in this term, and consequently, $\alpha_{\mathcal{S}_k}$ cannot appear in this term. Similarly, from Equation 4.18, $\Gamma_{\bar{\mathcal{S}}_k}$ depends on congestion factors $\alpha_{\mathcal{E}\cap\mathcal{C}_p}$, where if $p = q$, then $\mathcal{E} \cap \mathcal{C}_q \neq \mathcal{S}_k$, hence, the congestion factor $\alpha_{\mathcal{S}_k}$ cannot appear in this term as well.

In conclusion, all correlation subsets $\mathcal{S}_l \in S$ whose congestion factors $\alpha_{\mathcal{S}_l}$ appear in $\Gamma_{\mathcal{S}_k}$ or $\Gamma_{\bar{\mathcal{S}}_k}$, satisfy $\mathcal{S}_l \prec \mathcal{S}_k$. $\square$

Lemma 4.3. *The congestion probability of links set $\mathcal{E}$, i.e., $\mathbb{P}\left(\bigcap_{e_j\in\mathcal{E}} \{Z_{e_j} = 1\}\right)$, is uniquely determined from the congestion factors $\alpha_{\mathcal{S}_k}$ (Definition 4.1.6) of correlation subsets $\mathcal{S}_k \in S$, for all sets of links $\mathcal{E} \subseteq E$.*

Proof. As discussed in Section 4.5, the congestion probability of any set of links can be expressed as the product of the congestion probabilities of various correlation subsets. Therefore, it suffices to show that we can compute the congestion probabilities of all possible correlation subsets $\mathcal{S}_k \in S$ (Definition 4.0.5).

We prove our lemma for each correlation set $\mathcal{C}_p \in C$ (Definition 4.0.3). More precisely, we show that if we know the congestion factors $\alpha_{\mathcal{S}_k}$ of all correlation subsets $\mathcal{S}_k \subseteq \mathcal{C}_p$, we can determine the probability $\mathbb{P}\left(\cap_{e_j\in\mathcal{E}} \{Z_{e_j} = 1\}\right)$ for any set of links $\mathcal{E} \subseteq \mathcal{C}_p$.

First, we compute the probability of the event $\mathbf{S}_{\mathcal{C}_p} = \emptyset$ with $\mathbf{S}_{\mathcal{C}_p}$ given by Definition 4.1.3, i.e., all links in correlation set $\mathcal{C}_p$ are good:

$$\begin{aligned}\mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \emptyset\right) &= 1 - \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} \neq \emptyset\right) = 1 - \sum_{\mathcal{S}_k\subseteq\mathcal{C}_p, \mathcal{S}_k\neq\emptyset} \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \mathcal{S}_k\right) \\ &= 1 - \sum_{\mathcal{S}_k\subseteq\mathcal{C}_p, \mathcal{S}_k\neq\emptyset} \alpha_{\mathcal{S}_k}\, \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \emptyset\right).\end{aligned}$$

Since we know the congestion factors $\alpha_{\mathcal{S}_k}$ of all correlation subsets $\mathcal{S}_k \in \mathcal{C}_p$, we can compute $\mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \emptyset\right)$ from the above equation:

$$\mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \emptyset\right) = \frac{1}{1 + \sum_{\mathcal{S}_k\subseteq\mathcal{C}_p, \mathcal{S}_k\neq\emptyset} \alpha_{\mathcal{S}_k}}.$$

Using this result, we can compute the probability of the event $\mathbf{S}_{\mathcal{C}_p} = \mathcal{S}_k$, i.e., the links in $\mathcal{S}_k$ are the only congested links in correlation set $\mathcal{C}_p$, for all $\mathcal{S}_k \subseteq \mathcal{C}_p$:

$$\mathbb{P}\left(\ \mathbf{S}_{\mathcal{C}_p} = \mathcal{S}_k\ \right) = \alpha_{\mathcal{S}_k}\mathbb{P}\left(\ \mathbf{S}_{\mathcal{C}_p} = \emptyset\ \right) = \frac{\alpha_{\mathcal{S}_k}}{1 + \sum\limits_{\mathcal{S}_l \subseteq \mathcal{C}_p, \mathcal{S}_l \neq \emptyset} \alpha_{\mathcal{S}_l}}. \tag{4.19}$$

Finally, we determine the probability that all links in $\mathcal{E}$ are congested for any set of links $\mathcal{E} \subseteq \mathcal{C}_p$. The links in $\mathcal{E} \subseteq \mathcal{C}_p$ are simultaneously congested if and only if all links belonging to correlation subsets $\mathcal{S}_k \subseteq \mathcal{C}_p$ with $\mathcal{E} \subseteq \mathcal{S}_k$ are simultaneously congested. From the law of total probability and Equation 4.19, we obtain:

$$\mathbb{P}\left(\bigcap_{e_j \in \mathcal{E}} \{Z_{e_j} = 1\}\right) = \sum_{\substack{\mathcal{S}_k \subseteq \mathcal{C}_p \\ \text{s.t. } \mathcal{E} \subseteq \mathcal{S}_k}} \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \mathcal{S}_k\right) = \frac{\sum\limits_{\substack{\mathcal{S}_k \subseteq \mathcal{C}_p \\ \text{s.t. } \mathcal{E} \subseteq \mathcal{S}_k}} \alpha_{\mathcal{S}_k}}{1 + \sum\limits_{\mathcal{S}_l \subseteq \mathcal{C}_p, \mathcal{S}_l \neq \emptyset} \alpha_{\mathcal{S}_l}}.$$

□

Lemma 4.4. *The congestion factors $\alpha_{\mathcal{S}_k}$ (Definition 4.1.6) of all correlation subsets $\mathcal{S}_k \in S$, are uniquely determined from the congestion probabilities of link sets $\mathcal{E} \subseteq E$, i.e., $\mathbb{P}\left(\bigcap_{e_j \in \mathcal{E}} \{Z_{e_j} = 1\}\right)$.*

Proof. We prove the lemma for each correlation set $\mathcal{C}_p \in C$ (Definition 4.0.3). More precisely, we show that if we know the probability $\mathbb{P}\left(\ \cap_{e_j \in \mathcal{E}}\ \{Z_{e_j} = 1\}\right)$ for any set of links $\mathcal{E} \subseteq \mathcal{C}_p$, we can determine the congestion factors $\alpha_{\mathcal{S}_k}$ of all correlation subsets $\mathcal{S}_k \subseteq \mathcal{C}_p$.

We order all correlation subsets $\mathcal{S}_k \subseteq C_p$ in decreasing order of the number of links in each subset, that is, correlation subset $\mathcal{S}_l$ comes before correlation subset $\mathcal{S}_k$ in this ordering if $\mathcal{S}_l$ contains more links then $\mathcal{S}_k$. Following this order, we compute for each correlation subset $\mathcal{S}_k \subseteq \mathcal{C}_p$, the probability of the event $\mathbf{S}_{\mathcal{C}_p} = \mathcal{S}_k$ with $\mathbf{S}_{\mathcal{C}_p}$ given by Definition 4.1.3, i.e., the links in $\mathcal{S}_k$ are the only congested links in $\mathcal{C}_p$. By applying the law of total probability for the events that links in correlation subsets belonging to $\mathcal{C}_p$ are congested, we obtain:

$$\mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \mathcal{S}_k\right) = \mathbb{P}\left(\bigcap_{e_j \in \mathcal{S}_k} \{Z_{e_j} = 1\}\right) - \sum_{\substack{\mathcal{S}_l \subseteq \mathcal{C}_p \\ s.t.\ \mathcal{S}_k \subset \mathcal{S}_l}} \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \mathcal{S}_l\right).$$

Therefore, in order to compute $\mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \mathcal{S}_k\right)$, we need to know the probabilities $\mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \mathcal{S}_l\right)$, for which $\mathcal{S}_k \subset \mathcal{S}_l \subseteq \mathcal{C}_p$. The condition $\mathcal{S}_k \subset \mathcal{S}_l$ implies that correlation subset $\mathcal{S}_k$ contains fewer links than correlation subset $\mathcal{S}_l$. Hence, $\mathcal{S}_l$ comes before $\mathcal{S}_k$ in our ordering, which implies that we have already computed the probability $\mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \mathcal{S}_l\right)$ in a previous step.

Once we have determined the probabilities $\mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \mathcal{S}_k\right)$, for all $\mathcal{S}_k \subseteq \mathcal{C}_p$, we can compute the probability that all links in correlation set $\mathcal{C}_p$ are good:

$$\mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \emptyset\right) = 1 - \sum_{\substack{\mathcal{S}_k \subseteq \mathcal{C}_p \\ \mathcal{S}_k \neq \emptyset}} \mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \mathcal{S}_k\right).$$

Finally, we determine the congestion factors using Definition 4.1.6:

$$\alpha_{\mathcal{S}_k} = \frac{\mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \mathcal{S}_k\right)}{\mathbb{P}\left(\mathbf{S}_{\mathcal{C}_p} = \emptyset\right)},$$

for all correlation subsets $\mathcal{S}_k \subseteq \mathcal{C}_p$. □

Proof of Theorem 4.1

Proof. Identifying the congestion probabilities of all sets of links is equivalent with identifying the congestion factors of all correlation subsets given by Definition 4.1.6: If the congestion factors $\alpha_{\mathcal{S}_k}$ are known for all $\mathcal{S}_k \in S$, then we can compute the congestion probabilities of all links sets $\mathcal{E} \subseteq E$ (Lemma 4.3). On the other hand, if the congestion probability of link set $\mathcal{E}$ is known for all possible sets of links $\mathcal{E} \subseteq E$, then we can determine the congestion factors $\alpha_{\mathcal{S}_k}$, for all $\mathcal{S}_k \in S$ (Lemma 4.4). Therefore, in order to prove our theorem, it suffices to show that the congestion factors are identifiable from end-to-end measurements if and only if the Identifiability++ condition holds, i.e., no two correlation subsets are traversed by exactly the same paths.

The Identifiability++ condition is sufficient. First, we show that if the Identifiability++ condition holds, then all congestion factors are identifiable from end-to-end measurements. We prove this case by induction on the partial ordering $\mathcal{O}_S$ given by Definition 4.1.8.

Initial Step. Let $\mathcal{S}_1$ be the first element in the partial ordering $\mathcal{O}_S$. We will prove that we can compute the congestion factor $\alpha_{\mathcal{S}_1}$ from Equation 4.16.

First, we show by contradiction that $\mathcal{E} = \mathcal{S}_1$ is the only network state which satisfies $\mathit{Paths}(\mathcal{E}) = \mathit{Paths}(\mathcal{S}_1)$ with $\mathit{Paths}(\mathcal{E})$ the path coverage function given

by Definition 4.1.1. Suppose that there is another network state $\mathcal{E}^{'} \neq \mathcal{S}_1$, which satisfies $Paths(\mathcal{E}^{'}) = Paths(\mathcal{S}_1)$. We denote by $\mathcal{C}_q$ the correlation set of $\mathcal{S}_1$. We will show that (i) $\mathcal{E}^{'} \cap \mathcal{C}_p = \emptyset$, for all $\mathcal{C}_p \in C$, with $p \neq q$, and that (ii) $\mathcal{E}^{'} \cap \mathcal{C}_q = \mathcal{S}_1$.

Proposition (i). If $\mathcal{E}^{'} \cap \mathcal{C}_p \neq \emptyset$ for some correlation set $\mathcal{C}_p$, with $p \neq q$, then the congestion factor $\alpha_{\mathcal{E}^{'} \cap \mathcal{C}_p}$ must appear in either $\Gamma_{\mathcal{S}_1}$ given by Equation 4.17 (if $\mathcal{E}^{'} \cap \mathcal{C}_q = \mathcal{S}_1$) or in $\Gamma_{\bar{\mathcal{S}_1}}$ given by Equation 4.18 (if $\mathcal{E}^{'} \cap \mathcal{C}_q \neq \mathcal{S}_1$). From Lemma 4.2, we know that all congestion factors $\alpha_{\mathcal{E}^{'} \cap \mathcal{C}_p}$ which appear in $\Gamma_{\mathcal{S}_1}$ or $\Gamma_{\bar{\mathcal{S}_1}}$ must satisfy $(\mathcal{E}^{'} \cap \mathcal{C}_p) \prec \mathcal{S}_1$. Since correlation subset $\mathcal{S}_1$ is the first element in the ordering $\mathcal{O}_S$, there cannot be another correlation subset $\mathcal{E}^{'} \cap \mathcal{C}_p$ that comes before $\mathcal{S}_1$ in this ordering. Thus, we obtain that $\mathcal{E}^{'} \cap \mathcal{C}_p = \emptyset$ for all $\mathcal{C}_p \in C$, with $p \neq q$.

Proposition (ii). From Proposition (i), we know that $\mathcal{E}^{'} \cap \mathcal{C}_p = \emptyset$, for all $\mathcal{C}_p \in C$, with $p \neq q$, which implies that either $\mathcal{E}^{'} \neq \emptyset$ or $\mathcal{E}^{'} \subseteq \mathcal{C}_q$. Now, $\mathcal{E}^{'} \neq \emptyset$ because $Paths(\mathcal{E}^{'}) = Paths(\mathcal{S}_1) \neq \emptyset$, hence, $\mathcal{E}^{'}$ is a correlation subset. From the Identifiability++ condition, we know that no two correlation subsets are traversed by the same paths; thus, $\mathcal{E}^{'} = \mathcal{S}_1$.

Propositions (i) and (ii) imply that $\mathcal{E}^{'} = \mathcal{E} = \mathcal{S}_1$ is the only network state which satisfies $Paths(\mathcal{E}) = Paths(\mathcal{S}_1)$. Therefore, Equation 4.17 yields that:

$$\Gamma_{\mathcal{S}_1} = \sum_{\substack{\mathcal{E}=\mathcal{S}_1, \mathcal{E} \neq \mathcal{S}_1 \text{ s.t. } \mathcal{E} \cap \mathcal{C}_q = \mathcal{S}_1, \\ Paths(\mathcal{E}) = Paths(\mathcal{S}_1)}} \left(\prod_{\substack{\mathcal{C}_p \in C, p \neq q \\ \mathcal{E} \cap \mathcal{C}_p \neq \emptyset}} \alpha_{\mathcal{E} \cap \mathcal{C}_p} \right) = 0,$$

since $\mathcal{E} = \mathcal{S}_1$ contradicts the fact that $\mathcal{E} \neq \mathcal{S}_1$. Furthermore, Equation 4.18 yields that:

$$\Gamma_{\bar{\mathcal{S}_1}} = \sum_{\substack{\mathcal{E}=\mathcal{S}_1 \text{ s.t. } \mathcal{E} \cap \mathcal{C}_q \neq \mathcal{S}_1, \\ Paths(\mathcal{E}) = Paths(\mathcal{S}_1)}} \left(\prod_{\substack{\mathcal{C}_p \in C \\ \mathcal{E} \cap \mathcal{C}_p \neq \emptyset}} \alpha_{\mathcal{E} \cap \mathcal{C}_p} \right) = 0,$$

since $\mathcal{E} = \mathcal{S}_1$ and $\mathcal{S}_1 \subseteq \mathcal{C}_q$ contradict the fact that $\mathcal{E} \cap \mathcal{C}_q \neq \mathcal{S}_1$.

As a result, from Equation 4.16, we obtain:

$$\alpha_{\mathcal{S}_1} = \frac{\mathbb{P}\left(\; Paths(\mathbf{S}) = Paths(\mathcal{S}_1) \; \right)}{\mathbb{P}\left(\; Paths(\mathbf{S}) = \emptyset \; \right)}$$

where the term on the right is obtained through end-to-end measurements. Thus, we can compute $\alpha_{\mathcal{S}_1}$.

Induction Step. Consider a correlation subset $\mathcal{S}_k \in S$, and its correlation set $\mathcal{C}_q$, i.e., $\mathcal{S}_k \subseteq \mathcal{C}_q$. We assume that we know the congestion factors $\alpha_{\mathcal{S}_l}$ of all correlation subsets $\mathcal{S}_l \prec \mathcal{S}_k$. We will prove that we can compute $\alpha_{\mathcal{S}_k}$ from Equation 4.16.

First, we consider the case when $\mathbb{P}\left(Paths(\mathbf{S}) = Paths(\mathcal{S}_k)\right) = 0$. From Equation 4.13, we obtain $\mathbb{P}\left(\ \mathbf{S} = \mathcal{S}_k\ \right) = 0$, which implies that $\mathbb{P}\left(\ \mathbf{S}_{\mathcal{C}_q} = \mathcal{S}_k\ \right) = 0$, and consequently, $\alpha_{\mathcal{S}_k} = 0$.

Second, we consider the case when $\mathbb{P}\left(Paths(\mathbf{S}) = Paths(\mathcal{S}_k)\right) \neq 0$. According to Lemma 4.2, $\Gamma_{\mathcal{S}_k}$ and $\Gamma_{\bar{\mathcal{S}}_k}$ depend only on congestion factors $\alpha_{\mathcal{S}_l}$ of correlation subsets $\mathcal{S}_l \in S$, for which $\mathcal{S}_l \prec \mathcal{S}_k$. From the induction hypothesis, we know all these congestion factors; therefore, we can compute $\Gamma_{\mathcal{S}_k}$ and $\Gamma_{\bar{\mathcal{S}}_k}$ from Equations 4.17 and 4.18, and determine $\alpha_{\mathcal{S}_k}$ from Equation 4.16 as:

$$\alpha_{\mathcal{S}_k} = \frac{1}{1+\Gamma_{\mathcal{S}_k}}\left(\frac{\mathbb{P}\left(\ Paths(\mathbf{S}) = Paths(\mathcal{S}_k)\ \right)}{\mathbb{P}\left(\ Paths(\mathbf{S}) = \emptyset\ \right)} - \Gamma_{\bar{\mathcal{S}}_k}\right),$$

where the term on the right is obtained through end-to-end measurements.

The Identifiability++ condition is necessary. We will now show that if the Identifiability++ condition does not hold, then the congestion factors of all correlation subsets are not identifiable for all possible probability distributions of the congested links. More precisely, we prove that if there are two correlation subsets $\mathcal{S}_l$ and $\mathcal{S}_k$ such that $Paths(\mathcal{S}_l) = Paths(\mathcal{S}_k)$, then the congestion factors $\alpha_{\mathcal{S}_l}$ and $\alpha_{\mathcal{S}_k}$ are not identifiable from end-to-end measurements when the probability distribution of the congested links is such that:

$$\mathbb{P}\left(\mathbf{S} = \mathcal{E}\right) > 0 \text{ only if } \mathcal{E} \in \{\emptyset, \mathcal{S}_l, \mathcal{S}_k, \mathcal{S}_l \cup \mathcal{S}_k\}, \tag{4.20}$$

where $\mathbf{S}$ is the network state given by Definition 4.1.2.

In the context of Boolean loss tomography, we can learn from end-to-end measurements only the congested paths in each snapshot. Therefore, in order to compute the congestion factors of the correlation subsets, we can only use the probability that sets of paths are congested. For the probability distribution of the congested links in Equation 4.20, the possible network states and their outcome visible from end-to-end measurements is:

$\mathbf{S}$	Description	$Paths(\mathbf{S})$
$\mathcal{E}_1 = \emptyset$	all links are good	$Paths(\mathcal{E}_1) = \emptyset$
$\mathcal{E}_2 = \mathcal{S}_l$	only the links in $\mathcal{S}_l$ are congested	$Paths(\mathcal{E}_2) = Paths(\mathcal{S}_l)$
$\mathcal{E}_3 = \mathcal{S}_k$	only the links in $\mathcal{S}_k$ are congested	$Paths(\mathcal{E}_3) = Paths(\mathcal{S}_k)$
$\mathcal{E}_4 = \mathcal{S}_l \cup \mathcal{S}_k$	only the links in $\mathcal{S}_l \cup \mathcal{S}_k$ are congested	$Paths(\mathcal{E}_4) = Paths(\mathcal{S}_l \cup \mathcal{S}_k)$

From the hypothesis, we know that $Paths(\mathcal{S}_l) = Paths(\mathcal{S}_k) = Paths(\mathcal{S}_l \cup \mathcal{S}_k)$, hence, the probability distribution of the congested paths is such that:

$$\mathbb{P}\left(Paths(\mathbf{S}) = \mathcal{P}\right) > 0 \text{ only if } \mathcal{P} \in \{\emptyset, Paths(\mathcal{S}_l)\}. \tag{4.21}$$

We consider the probability of the event $(Paths(\mathbf{S}) = Paths(\mathcal{S}_l))$ (Definition 4.1.5), i.e., the only congested paths in the network are the paths that traverse links in correlation subset $\mathcal{S}_l$:

$$\begin{aligned}\mathbb{P}\left(Paths(\mathbf{S}) = Paths(\mathcal{S}_l)\right) &= \sum_{i=2}^{4} \mathbb{P}\left(\mathbf{S} = \mathcal{E}_i\right) \\ &= \mathbb{P}\left(\mathbf{S} = \mathcal{S}_l\right) + \mathbb{P}\left(\mathbf{S} = \mathcal{S}_k\right) + \mathbb{P}\left(\mathbf{S} = \mathcal{S}_l \cup \mathcal{S}_k\right).\end{aligned}$$

If we divide by Equation 4.11, we obtain:

$$\frac{\mathbb{P}\left(Paths(\mathbf{S}) = Paths(\mathcal{S}_l)\right)}{\mathbb{P}\left(Paths(\mathbf{S}) = \emptyset\right)} = \frac{\mathbb{P}\left(\mathbf{S} = \mathcal{S}_l\right)}{\mathbb{P}\left(\mathbf{S} = \emptyset\right)} + \frac{\mathbb{P}\left(\mathbf{S} = \mathcal{S}_k\right)}{\mathbb{P}\left(\mathbf{S} = \emptyset\right)} + \frac{\mathbb{P}\left(\mathbf{S} = \mathcal{S}_l \cup \mathcal{S}_k\right)}{\mathbb{P}\left(\mathbf{S} = \emptyset\right)}. \tag{4.22}$$

We distinguish two cases: (i) $\mathcal{S}_k$ and $\mathcal{S}_l$ belong to different correlation sets and (ii) $\mathcal{S}_k$ and $\mathcal{S}_l$ belong to the same correlation set. Because of Equation 4.9, in Case (i), Equation 4.22 becomes:

$$\frac{\mathbb{P}\left(Paths(\mathbf{S}) = Paths(\mathcal{S}_l)\right)}{\mathbb{P}\left(Paths(\mathbf{S}) = \emptyset\right)} = \alpha_{\mathcal{S}_l} + \alpha_{\mathcal{S}_k} + \alpha_{\mathcal{S}_l}\alpha_{\mathcal{S}_k}, \tag{4.23}$$

while in Case (ii), Equation 4.22 becomes:

$$\frac{\mathbb{P}\left(Paths(\mathbf{S}) = Paths(\mathcal{S}_l)\right)}{\mathbb{P}\left(Paths(\mathbf{S}) = \emptyset\right)} = \alpha_{\mathcal{S}_l} + \alpha_{\mathcal{S}_k} + \alpha_{\mathcal{S}_l \cup \mathcal{S}_k}. \tag{4.24}$$

From Equation 4.21, we know that $\mathbb{P}\left(Paths(\mathbf{S}) = \mathcal{P}\right) = \emptyset$, for all $\mathcal{P} \notin \{\emptyset, Paths(\mathcal{S}_k)\}$, and Equation 4.16 yields:

$$0 = \alpha_{\mathcal{S}_j}(1 + \Gamma_{\mathcal{S}_j}) + \Gamma_{\bar{\mathcal{S}}_j},$$

for any other correlation subset $\mathcal{S}_j \in S$, with $\mathcal{S}_j \notin \{\mathcal{S}_k, \mathcal{S}_l, \mathcal{S}_k \cup \mathcal{S}_l\}$. As a result, in both Case (i) and Case (ii), we cannot identify the congestion factors $\alpha_{\mathcal{S}_k}$ and $\alpha_{\mathcal{S}_l}$ using only Equation 4.23 or respectively, 4.24.

In conclusion, we have shown that the Identifiability++ condition is both necessary and sufficient in order to identify the congestion probabilities of all link sets. □

4.8 Conclusion

In this chapter, we have studied network loss tomography on correlated links. Previous works in network tomography implicitly assume that all links are independent. Nevertheless, there are practical scenarios in which links are correlated as they share physical links, network equipment, or even management processes. When such correlations occur in practice, the links' loss characteristics estimated by current tomographic algorithms might be inaccurate, moreover, there is no way of knowing to which extent they are inaccurate. We have proposed a model that takes into account correlated links. In particular, our model assumes that we know which links are most likely to be correlated (e.g., links from the same local area network or the same administrative domain), without assuming anything about the nature or the degree of their correlations (e.g., we do not assume knowledge of any correlation coefficients). For this correlation model, we have formally derived the necessary and sufficient condition under which it is feasible to identify the probability that all links in a set are congested, for all possible sets of links. Therefore, the congestion probability of each set of links can be obtained accurately under weaker assumptions than those required by state-of-the-art tomographic algorithms.

Symbol	Definition
$\mathcal{E} \subseteq E$	a set of links
$\mathcal{C}_p$	a correlation set
C	the set of all correlation sets
$\mathcal{S}_k$	a correlation subset
S	the set of all correlation subsets
$Paths(\mathcal{E})$	all paths traversing links in $\mathcal{E}$
$\mathbb{P}\left(Z_{e_j} = 1\right)$	the congestion probability of link e_j
$\mathbb{P}\left(\cap_{e_j \in \mathcal{E}} \{Z_{e_j} = 1\}\right)$	the congestion probability of link set $\mathcal{E}$
$\mathbb{P}\left(\cap_{e_j \in \mathcal{S}_k} \{Z_{e_j} = 1\}\right)$	the congestion probability of correlation subset $\mathcal{S}_k$
$\mathbf{S}$	the network state equal to the set of all congested links during a snapshot
$\mathbf{S}_{\mathcal{C}_p}$	the state of correlation set $\mathcal{C}_p$ equal to all congested links in $\mathcal{C}_p$ during a snapshot
$Paths(\mathbf{S})$	all congested paths in the network during a snapshot
$\mathbf{S} = \mathcal{E}$	the event that the links in $\mathcal{E}$ are the only congested links in the network
$\mathbf{S}_{\mathcal{C}_p} = \mathcal{S}_k$	the event that the links in correlation subset $\mathcal{S}_k \subseteq \mathcal{C}_p$ are the only congested links in correlation set $\mathcal{C}_p$
$Paths(\mathbf{S}) = Paths(\mathcal{S}_k)$	the event that the paths traversing links in correlation subset $\mathcal{S}_k$ are the only congested paths in the network
$\alpha_{\mathcal{S}_k}$	the congestion factor of correlation subset $\mathcal{S}_k$
$\mathcal{S}_k \prec \mathcal{S}_l$	the links in correlation subset $\mathcal{S}_k$ are traversed by fewer paths than the links in correlation subset $\mathcal{S}_l$
$\mathcal{O}_S$	a partial ordering of all correlation subsets in S induced by the precedence relation "$\prec$"

Table 4.1: Symbols defined in Chapter 4.

Chapter 5

A Different Loss Tomography

In this chapter, we argue for a different loss tomography: Congestion Probability Inference that computes the probability that each set of links is congested. Our motivation comes from the fact that both the continuous and Boolean loss tomography problems are ill-posed, that is, neither the loss rates, nor the congestion statuses of links are generally identifiable from end-to-end measurements. State-of-the-art tomographic algorithms try to counterbalance this with various assumptions as discussed in Section 2.5. Unfortunately, these assumptions do not usually hold in practice and can lead to inaccurate estimates of the loss characteristics of links. We do not attribute the blame to the limitations of particular tomographic algorithms, rather to the fundamental difficulty of solving the traditional versions of network loss tomography.

Congestion Probability Inference, however, is a well-posed problem under certain well-defined conditions: Under the assumption that all links are independent (the Link Independence assumption described in Section 4), this problem is well posed and there exists an algorithm that solves it [NT07a]. Under the assumption that links are grouped into known correlation sets (the Correlation Sets assumption described in Section 4.2), this problem is well-posed if and only if no two sets of potentially correlated links are traversed by the same paths, i.e., the Identifiability++ condition holds (Theorem 4.1). As our principle is to rely on the weakest set of assumptions possible, we study Congestion Probability Inference in the latter case, i.e., in the context of the link correlation model introduced in Section 4.2.

We model Congestion Probability Inference as a system of linear equations where each equation corresponds to a set of paths. Because it is not practically feasible to consider an equation for each set of paths in the network, we design

an algorithm that finds the maximum number of linearly independent equations by selecting particular sets of paths based on our theoretical results. On one hand, the information provided by our algorithm is less than that provided by the existing alternatives that infer either the loss rates or the congestion statuses of links, i.e., we only learn how often each set of links is congested, as opposed to how many packets were lost at each link, or which particular links were congested when. On the other hand, we show that this information is more useful in practice, because our algorithm works under the weakest set of assumptions to date, and we experimentally show that it is accurate under challenging network conditions such as non-stationary network dynamics and sparse topologies.

The rest of this chapter is organized as follows: We formally describe the problem of Congestion Probability Inference in Section 5.1, and we propose an algorithm which solves it in Section 5.2. We explain our choice of a different loss tomography with a practical scenario in Section 5.3. We compare Congestion Probability Inference with Boolean loss tomography in Sections 5.4 and 5.5, and we conclude in Section 5.6.

5.1 Congestion Probability Inference

In this section, we describe the problem of Congestion Probability Inference, whose goal is to determine, for each set of links $\mathcal{E} \subseteq E$, the probability that all links in $\mathcal{E}$ are congested. Similar to Boolean loss tomography, Congestion Probability Inference separates links and paths into *good* or *congested* such that a path is good if and only if all the links it traverses are good, i.e., the Separability assumption holds. Therefore, we use the same random variables defined in Section 2.3 to describe the congestion status of a link and that of a path: the random variable Z_{e_j} as the indicator of the congestion status of link e_j (Definition 2.0.1), and the random variable W_{p_i} as the indicator of the congestion status of path p_i (Definition 2.0.2). In this case, the quantities of interest are the congestion probabilities of sets of links defined in Section 4.5, that is, $\mathbb{P}\left(\cap_{e_j \in \mathcal{E}} \{Z_{e_j} = 1\}\right)$ for all set of links $\mathcal{E} \subseteq E$.

We have already formally shown that in the context of the link correlation model described in Section 4.2, the congestion probability of each set of links is identifiable from end-to-end measurements under certain well-defined conditions (Theorem 4.1). Moreover, the proof of Theorem 4.1, which is a proof by construction, describes an algorithm that solves Congestion Probability Infer-

Assumption	*Description*
Routing Stability	a fundamental assumption of network tomography
Stationarity	a fundamental assumption of multiple-snapshot algorithms
Separability	the assumption made by Boolean loss tomography
Correlation Sets	the assumption made by our link correlation model
Identifiability++ (necessary and sufficient condition)	no two correlation subsets are traversed by the same paths

Table 5.1: Assumptions and conditions required by Congestion Probability Inference.

ence. Unfortunately, this algorithm is impractical because it does not yield a linear system of equations that we can easily solve in practice. We now propose a practical algorithm which enables us to compute the congestion probabilities of sets of links.

5.1.1 Assumptions

All assumptions made by Congestion Probability Inference were already described in the previous chapters. For clarity, we summarize them here. Apart from the fundamental assumptions made by network tomography about the topology (the Routing Stability assumption), and the end-to-end measurements (the Stationarity assumption for multiple-snapshot algorithms), Congestion Probability Inference inherits two more assumptions from Boolean loss tomography and the link correlation model described in Section 4.2. In particular, Congestion Probability Inference relies on the Separability assumption introduced in Section 2.3, which states that a path is good if and only if all the links it traverses are good, and on the Correlation Sets assumption introduced in Section 4.2, which assumes that we know which links are most likely to be correlated. According to Theorem 4.1, under these assumptions, the congestion probabilities of sets of links are identifiable from end-to-end measurements, if and only if no two correlation subsets are traversed by the same paths (the Identifiability++ condition holds). We summarize all the assumptions required by Congestion Probability Inference in Table 5.1.

5.1.2 Problem Statement

We now establish a relationship between the quantities of interest and the information available from end-to-end measurements.

We model Congestion Probability Inference as a system of linear equations where each equation corresponds to a different set of paths $\mathcal{P} \subseteq P$, more precisely, to the probability that all paths in $\mathcal{P}$ are good, i.e., $\mathbb{P}\left(\cap_{p_i \in \mathcal{P}} \{W_{p_i} = 0\}\right)$. Each unknown in our system of equations corresponds to a correlation subset $\mathcal{S}_k \in S$ as given by Definition 4.0.5, more precisely, to the probability that all links in $\mathcal{S}_k$ are good, i.e., $\mathbb{P}\left(\cap_{e_j \in \mathcal{S}_k} \{Z_{e_j} = 0\}\right)$. As explained in Section 4.5, if we know the probability that all links in a correlation subset $\mathcal{S}_k$ are good, for all correlation subsets $\mathcal{S}_k \in S$, we can easily compute the congestion probabilities of all sets of links, and consequently, solve Congestion Probability Inference.

For a better illustration of our system of equations, we define the *link coverage* function $\mathit{Links}(\mathcal{P})$ which maps a set of paths $\mathcal{P} \subseteq P$ to the set of all links traversed by at least one of these paths.

Definition 5.0.1. *The link coverage function applied to path set $\mathcal{P} \subseteq P$ is:*

$$\mathit{Links}(\mathcal{P}) = \{\, e_j \in E \mid e_j \in p_i \text{ for some } p_i \in \mathcal{P}\}.$$

For example, in Figure 5.1, $\mathit{Links}(\{p_1\}) = \{e_1, e_3\}$, $\mathit{Links}(\{p_1, p_2\}) = \{e_1, e_3, e_4\}$. The link coverage function is not the dual of the path coverage function given by Definition 4.1.1, i.e.,

$$\mathcal{P} \subseteq \mathit{Paths}(\mathit{Links}(\mathcal{P})), \text{ for all } \mathcal{P} \subseteq P,$$

and

$$\mathcal{E} \subseteq \mathit{Links}(\mathit{Paths}(\mathcal{E})), \text{ for all } \mathcal{E} \subseteq E.$$

For example, in Figure 5.1,

$$\{p_1\} \subseteq \mathit{Paths}(\mathit{Links}(\{p_1\})) = \mathit{Paths}(\{e_1, e_3\}) = \{p_1, p_2\},$$

and

$$\{e_1\} \subseteq \mathit{Links}(\mathit{Paths}(\{e_1\})) = \mathit{Links}(\{p_1, p_2\}) = \{e_1, e_3, e_4\}.$$

Consider the scenario that all paths in a set $\mathcal{P} \subseteq P$ are good. By the Separability assumption, this implies that all links traversed by these paths, i.e., the links in $\mathit{Links}(\mathcal{P})$, are good. Therefore, we can write:

$$\mathbb{P}\left(\bigcap_{p_i \in \mathcal{P}} \{W_{p_i} = 0\}\right) = \mathbb{P}\left(\bigcap_{e_j \in \mathit{Links}(\mathcal{P})} \{Z_{e_j} = 0\}\right).$$

If the Correlation Sets assumption holds, then links belonging to different correlation sets are independent; hence, we can group links by correlation sets (Definition 4.0.3):

$$\mathbb{P}\left(\bigcap_{p_i \in \mathcal{P}} \{W_{p_i} = 0\}\right) = \prod_{\mathcal{C}_p \in C} \mathbb{P}\left(\bigcap_{e_j \in Links(\mathcal{P}) \cap \mathcal{C}_p} \{Z_{e_j} = 0\}\right). \tag{5.1}$$

A set of links $Links(\mathcal{P}) \cap \mathcal{C}_p$, with $\mathcal{P} \subseteq P$ and $\mathcal{C}_p \in C$, is either the empty set if none of the links traversed by paths in $\mathcal{P}$ belongs to $\mathcal{C}_p$, or a correlation subset because it belongs to correlation set $\mathcal{C}_p$. In Figure 5.1, if we apply Equation 5.1 to path set $\{p_1\}$, we obtain:

$$\begin{aligned}\mathbb{P}(W_{p_1} = 0) &= \mathbb{P}(Z_{e_1} = 0, Z_{e_3} = 0) \\ &= \mathbb{P}(Z_{e_1} = 0)\ \mathbb{P}(Z_{e_3} = 0).\end{aligned}$$

Similarly, if we apply Equation 5.1 to path set $\{p_1, p_2\}$, we obtain:

$$\begin{aligned}\mathbb{P}(W_{p_1} = 0, W_{p_2} = 0) &= \mathbb{P}(Z_{e_1} = 0, Z_{e_3} = 0, Z_{e_4} = 0) \\ &= \mathbb{P}(Z_{e_1} = 0)\ \mathbb{P}(Z_{e_3} = 0, Z_{e_4} = 0).\end{aligned}$$

Equation 5.1 establishes a relationship between the probability that sets of paths are good and the probabilities that links belonging to various correlation subsets are good. Furthermore, if we take the logarithm of Equation 5.1, we obtain a linear equation. The probability that all paths in $\mathcal{P}$ are good can be measured directly from end-to-end measurements, for all sets of paths $\mathcal{P} \subseteq P$, while the probabilities that all links belonging to various correlation subsets are good represent the unknowns.

5.2 A Congestion Probability Inference Algorithm

In this section, we design an algorithm which solves Congestion Probability Inference, that is, it infers the congestion probabilities of sets of links from end-to-end measurements. Our algorithm does not make any additional assumptions other than the basic assumptions of Congestion Probability Inference summarized in Table 5.1.

In order to solve Congestion Probability Inference, a straightforward approach is to apply Equation 5.1 to all possible sets of paths in the network, to reduce this to a system of linearly independent equations, and to solve the

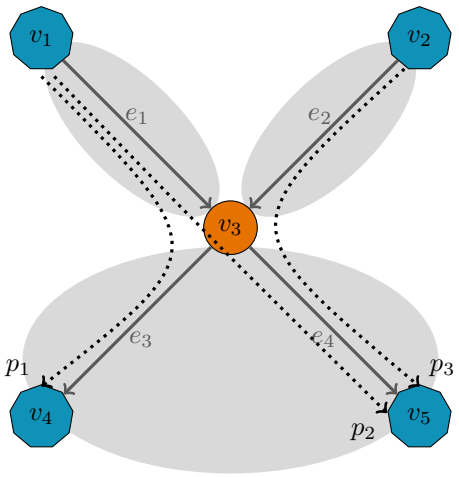


Figure 5.1: A toy topology with correlated links where the Identifiability++ condition holds, i.e., no two correlation subsets are traversed by the same paths. Hosts $V^H = \{v_1, v_2, v_4, v_5\}$. Routers $V^R = \{v_3\}$. Links $E = \{e_1, e_2, e_3, e_4\}$. Paths $P = \{p_1, p_2, p_3\}$. Correlation sets $\mathcal{C}_1 = \{e_1\}$, $\mathcal{C}_2 = \{e_2\}$, and $\mathcal{C}_3 = \{e_3, e_4\}$. Correlation subsets $S = \{\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_3, e_4\}\}$. We consider two scenarios: (Scenario A) path p_1 is good in all snapshots, whereas paths p_2 and p_3 are congested in at least one snapshot, and (Scenario B) all three paths are congested in at least one snapshot.

latter. However, there are $2^{|P|}$ possible sets of paths in the network, and processing $2^{|P|}$ equations is practically infeasible for any topology with more than a few tens of paths. Under the Link Independence assumption, the algorithm in [NT07a] tries to determine the congestion probabilities of individual links by applying Equation 5.1 only to paths and to pairs of paths. However, in this case, there is no guarantee that the resulting system of equations is not undetermined. We address this challenge by using a novel technique that under the Correlation Sets assumption, forms the maximum number of linearly independent equations possible, without applying Equation 5.1 to all sets of paths.

5.2.1 Definitions and Notations

First, we introduce some basic definitions and notations. We illustrate the defined symbols in Table 5.2 by considering two possible scenarios for the toy topology in Figure 5.1. In Scenario A, from end-to-end measurements, we see that path p_1 is good in all snapshots, whereas paths p_2 and p_3 are congested in at least one snapshot. In Scenario B, from end-to-end measurements, we see

that all three paths are congested in at least one snapshot. All our symbols are summarized in Table 5.5.

The Potentially Congested Links

Since Boolean loss tomography is ill-posed, we cannot tell which links are congested during a snapshot just by looking at the congested paths during that snapshot. Nevertheless, for some of the links, the congestion status is visible from end-to-end measurements: If during a snapshot, at least one of the paths traversing link e_j is good, then by the Separability assumption, link e_j is also good during that snapshot. Furthermore, if in *each* snapshot of the experiment, one of the paths traversing link e_j is good, then link e_j remains good throughout *all* snapshots, and consequently, its congestion probability is zero, i.e., $\mathbb{P}\left(Z_{e_j} = 1\right) = 0$. In this case, we say that link e_j is *almost surely good* as end-to-end measurements ensure it is good. We refer to a link which is not almost surely good as *potentially congested* since its congestion probability maybe different than zero.

Definition 5.0.2. *$\widehat{E}$ is the set of all potentially congested links in the network.*

Therefore, all links which are not in $\widehat{E}$ have zero congestion probability. In Figure 5.1, Scenario A, links e_1 and e_3 are almost surely good, while links e_2 and e_4 are potentially congested, i.e., $\widehat{E} = \{e_2, e_4\}$. In Scenario B, all links are potentially congested, i.e., $\widehat{E} = \{e_1, e_2, e_3, e_4\}$.

A Potentially Congested Correlation Subset

Definition 5.0.3. *A correlation subset $\mathcal{S}_k \in S$ (Definition 4.0.5) is potentially congested if $\mathcal{S}_k \subseteq \widehat{E}$.*

Therefore, a *potentially congested correlation subset* $\mathcal{S}_k$ satisfies two conditions: (i) $\mathcal{S}_k$ is a non-empty subset of some correlation set $\mathcal{C}_p$, i.e., $\mathcal{S}_k \subseteq \mathcal{C}_p$ with $\mathcal{S}_k \neq \emptyset$, and (ii) all links in $\mathcal{S}_k$ are potentially congested, i.e., $\mathcal{S}_k \subseteq \widehat{E}$.

Definition 5.0.4. *$\widehat{S}$ is <u>any</u> ordering of all potentially congested correlation subsets.*

Note that we do not impose a specific order on $\widehat{S}$, the only requirement is that it contains all potentially congested correlation subsets. In Figure 5.1, Scenario A, there are two possible orderings of the potentially congested correlation subsets $\widehat{S} = \langle\{e_2\}, \{e_4\}\rangle$ and $\widehat{S} = \langle\{e_4\}, \{e_2\}\rangle$. In Scenario B, one of the possible orderings is $\widehat{S} = \langle\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_3, e_4\}\rangle$.

Next, we define the notion of the complement of a potentially congested correlation subset:

Definition 5.0.5. *The complement of $\mathcal{S}_k \in \widehat{S}$ (Definition 4.0.5) is*

$$\overline{\mathcal{S}}_k = (\mathcal{C}_p \setminus \mathcal{S}_k) \cap \widehat{E},$$

where $\mathcal{C}_p$ is the correlation set of $\mathcal{S}_k$.

Hence, $\overline{\mathcal{S}}_k$ consists of all potentially congested links in correlation set $\mathcal{C}_p$ excluding the links in $\mathcal{S}_k$. In Figure 5.1, Scenario A, $\overline{\{e_2\}} = \emptyset$ and $\overline{\{e_4\}} = \emptyset$, while in Scenario B, $\overline{\{e_1\}} = \emptyset$, $\overline{\{e_2\}} = \emptyset$, $\overline{\{e_3\}} = \{e_4\}$, $\overline{\{e_4\}} = \{e_3\}$, and $\overline{\{e_3, e_4\}} = \emptyset$.

The Potentially Congested Link Coverage Function

The *potentially congested link coverage* function $\widehat{Links}(\mathcal{P})$ maps a set of paths $\mathcal{P}$ to the set of all potentially congested links traversed by at least one of these paths.

Definition 5.0.6. *The potentially congested link coverage function applied to paths set $\mathcal{P} \subseteq P$ is:*

$$\widehat{Links}(\mathcal{P}) = \{\ e \in \widehat{E} \mid e \in p \text{ for some } p \in \mathcal{P}\}.$$

If we consider the link coverage function given by Definition 5.0.1, then

$$\widehat{Links}(\mathcal{P}) = Links(\mathcal{P}) \cap \widehat{E}. \tag{5.2}$$

In Figure 5.1, Scenario B, $\widehat{Links}(\{p_1, p_2\}) = \{e_1, e_3, e_4\}$, and $\widehat{Links}(\{p_2, p_3\}) = \{e_1, e_2, e_4\}$, whereas in Scenario A, $\widehat{Links}(\{p_1, p_2\}) = \{e_4\}$ and $\widehat{Links}(\{p_2, p_3\}) = \{e_2, e_4\}$ as links e_1 and e_3 are almost surely good.

By definition, the potentially congested link coverage function has the following properties: for any path sets $\mathcal{P}, \mathcal{Q} \subseteq P$,

$$\mathcal{Q} \subseteq \mathcal{P} \ \Rightarrow\ \widehat{Links}(\mathcal{Q}) \subseteq \widehat{Links}(\mathcal{P}), \tag{5.3}$$

$$\widehat{Links}(\mathcal{Q} \cup \mathcal{P}) = \widehat{Links}(\mathcal{Q}) \cup \widehat{Links}(\mathcal{P}). \tag{5.4}$$

For example, in Figure 5.1, Scenario B:

$$\widehat{Links}(\{p_1\}) = \{e_1, e_3\} \subseteq \{e_1, e_3, e_4\} = \widehat{Links}(\{p_1, p_2\}),$$

	Scenario A	Scenario B
Description	only path p_1 is good in all snapshots	each path is congested in at least one snapshot
$\widehat{E}$	$\{e_2, e_4\}$	$\{e_1, e_2, e_3, e_4\}$
$\widehat{S}$	$\widehat{S} = \langle\{e_2\}, \{e_4\}\rangle$	$\widehat{S} = \langle\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_3, e_4\}\rangle$
$\bar{\mathcal{S}_k},\ \mathcal{S}_k \in \widehat{S}$	$\overline{\{e_2\}} = \emptyset$, $\overline{\{e_4\}} = \emptyset$	$\overline{\{e_1\}} = \emptyset$, $\overline{\{e_2\}} = \emptyset$, $\overline{\{e_3\}} = \{e_4\}$, $\overline{\{e_4\}} = \{e_3\}$, $\overline{\{e_3, e_4\}} = \emptyset$
$\widehat{Links}(\mathcal{P})$	$\widehat{Links}(\{p_1, p_2\}) = \{e_4\}$ $\widehat{Links}(\{p_2, p_3\}) = \{e_2, e_4\}$	$\widehat{Links}(\{p_1, p_2\}) = \{e_1, e_3, e_4\}$ $\widehat{Links}(\{p_2, p_3\}) = \{e_1, e_2, e_4\}$

Table 5.2: An illustration of the symbols defined in Section 5.2.1 in two possible scenarios for the topology in Figure 5.1.

$$\widehat{Links}(\{p_1, p_2\}) = \{e_1, e_3, e_4\} = \{e_1, e_3\} \cup \{e_1, e_4\} = \widehat{Links}(\{p_1\}) \cup \widehat{Links}(\{p_2\}).$$

5.2.2 The System of Equations

The unknowns in Equation 5.1 are the probabilities that all links in a correlation subset $\mathcal{S}_k$ are good, for all $\mathcal{S}_k \in S$. However, for some of the correlation subsets, we already know these probabilities: If correlation subset $\mathcal{S}_k$ contains only almost surely good links, i.e., if $\mathcal{S}_k \cap \widehat{E} = \emptyset$ with $\widehat{E}$ given by Definition 5.0.2, then the probability that all links in $\mathcal{S}_k$ are good is equal to 1. Furthermore, if we are only interested in the probability that all links in any correlation subset $\mathcal{S}_k \in S$ are good, we can ignore the almost surely good links in $\mathcal{S}_k$ because these links remain good throughout all snapshots. Thus,

$$\mathbb{P}\left(\bigcap_{e_j \in \mathcal{S}_k} \{Z_{e_j} = 0\}\right) = \mathbb{P}\left(\bigcap_{e_j \in \mathcal{S}_k \cap \widehat{E}} \{Z_{e_j} = 0\}\right), \text{ for all } \mathcal{S}_k \in S. \tag{5.5}$$

Note that for any correlation subset $\mathcal{S}_k \in S$, the term $\mathcal{S}_k \cap \widehat{E}$ is either the empty set when all links in $\mathcal{S}_k$ are almost surely good, or a potentially congested correlation subset, and in this case, $(\mathcal{S}_k \cap \widehat{E}) \in \widehat{S}$ with $\widehat{S}$ given by Definition 5.0.4. Therefore, in order to solve Congestion Probability Inference, it suffices to know the probabilities that all links belonging to potentially congested correlation subsets in $\widehat{S}$ are good.

If we use Equation 5.5 in Equation 5.1, we obtain:

$$\mathbb{P}\left(\bigcap_{p_i\in\mathcal{P}}\{W_{p_i}=0\}\right)=\prod_{\mathcal{C}_p\in C}\mathbb{P}\left(\bigcap_{e_j\in(Links(\mathcal{P})\cap\mathcal{C}_p)\cap\widehat{E}}\{Z_{e_j}=0\}\right)$$
$$=\prod_{\mathcal{C}_p\in C}\mathbb{P}\left(\bigcap_{e_j\in\widehat{Links}(\mathcal{P})\cap\mathcal{C}_p}\{Z_{e_j}=0\}\right), \tag{5.6}$$

where $\widehat{Links}(\mathcal{P})$ is the potentially congested link coverage function given by Definition 5.0.6.

In Figure 5.1, Scenario B, since all paths are congested in at least one snapshot, then all links are potentially congested, and Equation 5.6 applied to path set $\{p_1, p_2\}$ reads:

$$\mathbb{P}\left(W_{p_1}=0, W_{p_2}=0\right)=\mathbb{P}\left(Z_{e_1}=0\right)\mathbb{P}\left(Z_{e_3}=0, Z_{e_4}=0\right).$$

In Figure 5.1, Scenario A, since path p_1 is good in all snapshots, then only links e_2 and e_4 are potentially congested, whereas e_1 and e_3 are almost surely good, and Equation 5.6 applied to path set $\{p_1, p_2\}$ reads:

$$\mathbb{P}\left(W_{p_1}=0, W_{p_2}=0\right)=\mathbb{P}\left(Z_{e_4}=0\right).$$

Therefore, in Scenario A, we can determine the probability that link e_4 is good as it is equal to the probability that paths p_1 and p_2 are both good.

Note that if we take the logarithm of Equation 5.6, we obtain a linear equation:

$$\log\mathbb{P}\left(\bigcap_{p_i\in\mathcal{P}}\{W_{p_i}=0\}\right)=\sum_{\mathcal{C}_p\in C}\log\mathbb{P}\left(\bigcap_{e_j\in\widehat{Links}(\mathcal{P})\cap\mathcal{C}_p}\{Z_{e_j}=0\}\right). \tag{5.7}$$

We denote by $\widehat{\mathcal{P}}$ an ordering of path sets (it does not necessarily include all possible path sets). We apply Equation 5.7 to each path set in $\widehat{\mathcal{P}}$, and we recast these equations in a vector form:

$$\mathbf{V}=Matrix(\widehat{\mathcal{P}},\widehat{S})\,\mathbf{U} \tag{5.8}$$

where $\widehat{S}$ is any ordering of the potentially congested correlation subsets,

$$\mathbf{V} = \left[\log \mathbb{P}\left(\cap_{p_i \in \mathcal{P}} \{W_{p_i} = 0\}\right)\right]_{\mathcal{P} \in \widehat{\mathcal{P}}} \tag{5.9}$$

is the vector of available measurements,

$$\mathbf{U} = \left[\log \mathbb{P}\left(\cap_{e_j \in \mathcal{S}_k} \{Z_{e_j} = 0\}\right)\right]_{\mathcal{S}_k \in \widehat{S}} \tag{5.10}$$

is the vector of unknowns, and $\mathit{Matrix}(\widehat{\mathcal{P}}, \widehat{S})$ is the matrix associated to the system of equations. Each row in $\mathit{Matrix}(\widehat{\mathcal{P}}, \widehat{S})$ corresponds to a path set $\mathcal{P} \in \widehat{\mathcal{P}}$, whereas each column corresponds to a potentially congested correlation subset $\mathcal{S}_k \in \widehat{S}$.

In Figure 5.1, Scenario B, suppose that $\widehat{S} = \langle \{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_3, e_4\} \rangle$. if we take $\widehat{\mathcal{P}} = \{\{p_1\}, \{p_1, p_2\}\}$ and we apply Equation 5.7 to each path set in $\widehat{\mathcal{P}}$, we obtain:

$$\log \mathbb{P}\left(W_{p_1} = 0\right) = \log \mathbb{P}\left(Z_{e_1} = 0\right) + \log \mathbb{P}\left(Z_{e_3} = 0\right),$$
$$\log \mathbb{P}\left(W_{p_1} = 0, W_{p_2} = 0\right) = \log \mathbb{P}\left(Z_{e_1} = 0\right) + \log \mathbb{P}\left(Z_{e_3} = 0, Z_{e_4} = 0\right).$$

We recast these equations in vector form as described by Equation 5.8, and we obtain:

$$\underbrace{\begin{bmatrix} \log \mathbb{P}\left(W_{p_1} = 0\right) \\ \log \mathbb{P}\left(W_{p_1} = 0, W_{p_2} = 0\right) \end{bmatrix}}_{\mathbf{V}} = \underbrace{\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathit{Matrix}(\widehat{\mathcal{P}}, \widehat{S})} \underbrace{\begin{bmatrix} \log \mathbb{P}\left(Z_{e_1} = 0\right) \\ \log \mathbb{P}\left(Z_{e_2} = 0\right) \\ \log \mathbb{P}\left(Z_{e_3} = 0\right) \\ \log \mathbb{P}\left(Z_{e_4} = 0\right) \\ \log \mathbb{P}\left(Z_{e_3} = 0, Z_{e_4} = 0\right) \end{bmatrix}}_{\mathbf{U}}. \tag{5.11}$$

In our analysis, we will often refer to the following terms:

Definition 5.0.7. *$\mathit{Row}(\mathcal{P}, \widehat{S})$ is the row in $\mathit{Matrix}(\widehat{\mathcal{P}}, \widehat{S})$ (Equation 5.8) corresponding to path set $\mathcal{P} \in \widehat{\mathcal{P}}$.*

Therefore, if $\mathcal{P}$ is the i-th path set in the ordering $\widehat{\mathcal{P}}$, then the i-th row of $\mathit{Matrix}(\widehat{\mathcal{P}}, \widehat{S})$ is $\mathit{Row}(\mathcal{P}, \widehat{S})$. For example, in the matrix in Equation 5.11, the first row is $\mathit{Row}(\{p_1\}, \widehat{S})$, whereas the second row is $\mathit{Row}(\{p_1, p_2\}, \widehat{S})$.

Definition 5.0.8. *The **coverage indicator** $\boldsymbol{\alpha}_{\mathcal{P},\mathcal{S}_k}$ is the element of Matrix$(\widehat{\mathcal{P}}, \widehat{S})$ (Equation 5.8) at the intersection of the row of path set $\mathcal{P} \in \widehat{\mathcal{P}}$ and of the column of correlation subset $\mathcal{S}_k \in \widehat{S}$, i.e.,*

$$\boldsymbol{\alpha}_{\mathcal{P},\mathcal{S}_k} = \begin{cases} 1 & \text{if } \widehat{Links}(\mathcal{P}) \cap \mathcal{C}_p = \mathcal{S}_k \\ 0 & \text{otherwise} \end{cases},$$

where $\mathcal{C}_p$ is the correlation set of $\mathcal{S}_k$.

In Figure 5.1, Scenario B, in the example in Equation 5.11, $\boldsymbol{\alpha}_{\{p_1\},\{e_3\}} = 1$ because $\widehat{Links}(\{p_1\}) \cap \{e_3, e_4\} = \{e_3\}$, but $\boldsymbol{\alpha}_{\{p_1,p_2\},\{e_3\}} = 0$ because $\widehat{Links}(\{p_1, p_2\}) \cap \{e_3, e_4\} = \{e_3, e_4\} \neq \{e_3\}$.

5.2.3 Theoretical Results

In this section, we formally show that in order to solve Congestion Probability Inference, we do not need to apply Equation 5.7 to all possible set of paths in the network. More specifically, we identify a particular type of paths sets, which we call *redundant*, and which have the following property: Equation 5.7 applied to a redundant path set is a linear combination of Equation 5.7 applied to non-redundant path sets. As a result, we can discard all redundant path sets which represent a big fraction of all path sets and apply Equation 5.7 only to non-redundant path sets.

A Redundant Path Set

Definition 5.0.9. *A path set $\mathcal{P}$ is called redundant if there is no potentially congested correlation subset $\mathcal{S}_k \in \widehat{S}$ such that $\mathcal{P} \subseteq Paths(\mathcal{S}_k) \setminus Paths(\overline{\mathcal{S}}_k)$, with $\overline{\mathcal{S}}_k$ the complement of $\mathcal{S}_k$ given by Definition 5.0.5, and $\widehat{S}$ by Definition 5.0.4.*

In the particular case where all links are independent, i.e., when the Link Independence assumption holds, Definition 5.0.9 yields that a path set $\mathcal{P}$ is redundant if there is no potentially congested link which is shared by all paths in $\mathcal{P}$. For example, in the toy topology in Figure 5.1, suppose that all links are independent and potentially congested. This setup is described in Table 5.3. In this case, path set $\{p_1, p_3\}$ is redundant since there is no potentially congested link which is traversed by both paths p_1 and p_3.

$\mathcal{S}_k \in \widehat{S}$	$\overline{\mathcal{S}}_k$	$Paths(\mathcal{S}_k)$	$Paths(\overline{\mathcal{S}}_k)$	$Paths(\mathcal{S}_k) \setminus Paths(\overline{\mathcal{S}}_k)$
$\{e_1\}$	$\emptyset$	$\{p_1, p_2\}$	$\emptyset$	$\{p_1, p_2\}$
$\{e_2\}$	$\emptyset$	$\{p_3\}$	$\emptyset$	$\{p_3\}$
$\{e_3\}$	$\emptyset$	$\{p_1\}$	$\emptyset$	$\{p_1\}$
$\{e_4\}$	$\emptyset$	$\{p_2, p_3\}$	$\emptyset$	$\{p_2, p_3\}$

Table 5.3: The scenario when all links in the toy topology in Figure 5.1 are independent and potentially congested. Path set $\{p_1, p_3\}$ is redundant since for all $\mathcal{S}_k$ in $\widehat{S}$, $\{p_1, p_3\} \not\subseteq Paths(\mathcal{S}_k) \setminus Paths(\overline{\mathcal{S}}_k)$.

Theorem 5.1 states that Equation 5.7 applied to a redundant path set $\mathcal{P}$ is a linear combination of the same equation (5.7) applied to path sets $\mathcal{Q} \subset \mathcal{P}$. We can already see this in the example described in Table 5.3. Indeed,

$$\log \mathbb{P}\left(W_{p_1} = 0, W_{p_3} = 0\right) = \log \mathbb{P}\left(W_{p_1} = 0\right) + \log \mathbb{P}\left(W_{p_3} = 0\right), \tag{5.12}$$

that is, Equation 5.7 applied to path set $\{p_1, p_3\}$ is the sum of Equation 5.7 applied respectively to path sets $\{p_1\}$ and $\{p_3\}$. Using Definition 5.0.7, Equation 5.12 implies that:

$$Row(\{p_1, p_3\}, \widehat{S}) = Row(\{p_1\}, \widehat{S}) + Row(\{p_3\}, \widehat{S}).$$

The Partition of a Path Set.

Consider a path set $\mathcal{P} \subseteq P$, and a potentially congested correlation subset $\mathcal{S}_k \in \widehat{S}$ (Definition 5.0.4). We denote by $\mathcal{C}_p$ the correlation set of $\mathcal{S}_k$, i.e., $\mathcal{S}_k \subseteq \mathcal{C}_p$, and by $\overline{\mathcal{S}}_k$ the complement of $\mathcal{S}_k$ given by Definition 5.0.5. We partition $\mathcal{P}$ into the three following path sets:

$$\mathcal{P}_{\overline{\mathcal{C}}_p} = \{\ p \in \mathcal{P} \mid \widehat{Links}(\{p\}) \cap \mathcal{C}_p = \emptyset\ \} \tag{5.13}$$

$$\mathcal{P}_{\overline{\mathcal{S}}_k} = \{\ p \in \mathcal{P} \mid \widehat{Links}(\{p\}) \cap \overline{\mathcal{S}}_k \neq \emptyset\ \} \tag{5.14}$$

$$\mathcal{P}_{\mathcal{S}_k} = Paths(\mathcal{S}_k) \setminus Paths(\overline{\mathcal{S}}_k). \tag{5.15}$$

The set $\mathcal{P}_{\overline{\mathcal{C}}_p}$ contains the paths in $\mathcal{P}$ which do not traverse any potentially congested link in correlation set $\mathcal{C}_p$ (Equation 5.13). The set $\mathcal{P}_{\overline{\mathcal{S}}_k}$ contains the paths in $\mathcal{P}$ which traverse at least one link in $\overline{\mathcal{S}}_k$ (Equation 5.14). Finally, the set $\mathcal{P}_{\mathcal{S}_k}$ contains the paths in $\mathcal{P}$ which traverse at least one link in $\mathcal{S}_k$, but do not traverse any link in $\overline{\mathcal{S}}_k$ (Equation 5.15).

Definition 5.0.10. *For a path set $\mathcal{P} \subseteq P$, and a correlation subset $\mathcal{S}_k \in \widehat{S}$, $\Omega_{\mathcal{P},\mathcal{S}_k}$ is defined as:*

$$\Omega_{\mathcal{P},\mathcal{S}_k} = \{\ \mathcal{Q} \subseteq \mathcal{P}_{\mathcal{S}_k} \mid \boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k} = 1\ \}.$$

where $\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k}$ is the coverage indicator given by Definition 5.0.8, and $\mathcal{P}_{\mathcal{S}_k}$ is the path set defined by Equation 5.15.

In Figure 5.1, Scenario B, consider path set $\mathcal{P} = \{p_1, p_2, p_3\}$ and the potentially congested correlation subset $\mathcal{S}_k = \{e_4\}$. In this case, $\mathcal{C}_p = \mathcal{C}_3 = \{e_3, e_4\}$ and $\overline{\mathcal{S}}_k = \overline{\{e_4\}} = \{e_3\}$. Therefore, we obtain $\mathcal{P}_{\overline{\mathcal{C}}_p} = \emptyset$ since all three paths traverse one of the links in correlation set $\mathcal{C}_3$, $\mathcal{P}_{\overline{\mathcal{S}}_k} = \{p_1\}$ since path p_1 is the only path which traverses link e_3, and $\mathcal{P}_{\mathcal{S}_k} = \{p_2, p_3\}$. Finally, we get $\Omega_{\mathcal{P},\mathcal{S}_k} = \{\{p_2\}, \{p_3\}, \{p_2, p_3\}\}$.

The Core Theorem

In this section, we formally show that Equation 5.7 applied to a redundant path set $\mathcal{P}$ is a linear combination of Equation 5.7 applied to all path sets $\mathcal{Q} \subset \mathcal{P}$. More specifically, in the context of the system described in Equation 5.8, we prove that the vectors $Row(\mathcal{Q}, \widehat{S})$ given by Definition 5.0.7, with $\mathcal{Q} \subseteq \mathcal{P}$, form a linearly dependent set for any ordering of the potentially congested correlation subsets $\widehat{S}$.

Theorem 5.1. *A redundant path set $\mathcal{P} \subseteq P$ satisfies*

$$\sum_{\mathcal{Q} \subseteq \mathcal{P}} (-1)^{|\mathcal{Q}|} Row(\mathcal{Q}, \widehat{S}) = \mathbf{0}, \tag{5.16}$$

where $\widehat{S}$ is any ordering of the potentially congested correlation subsets, and $Row(\mathcal{Q}, \widehat{S})$ is given by Definition 5.0.7.

Proof. In order to prove that Equation 5.16 holds, it is sufficient to show that, for any correlation subset $\mathcal{S}_k$ in $\widehat{S}$,

$$\sum_{\mathcal{Q} \subseteq \mathcal{P}} (-1)^{|\mathcal{Q}|} \boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k} = 0, \tag{5.17}$$

where $\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k}$ is the coverage indicator of path set $\mathcal{Q}$ and correlation subset $\mathcal{S}_k$ as given by Definition 5.0.8. For convenience, we introduce the following notation

$$\Gamma_{\mathcal{P},\mathcal{S}_k} = \sum_{\mathcal{Q}\subseteq\mathcal{P}} (-1)^{|\mathcal{Q}|}\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k}.$$

Since $\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k}$ only takes values 1 or 0, we can rewrite $\Gamma_{\mathcal{P},\mathcal{S}_k}$ as

$$\Gamma_{\mathcal{P},\mathcal{S}_k} = \sum_{\substack{\mathcal{Q}\subseteq\mathcal{P} \\ \text{s.t. } \boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k}=1}} (-1)^{|\mathcal{Q}|}. \tag{5.18}$$

We will prove that $\Gamma_{\mathcal{P},\mathcal{S}_k} = 0$, for all potentially congested correlation subsets $\mathcal{S}_k \in \widehat{S}$.

Consider a potentially congested correlation subset $\mathcal{S}_k \in \widehat{S}$, and the correlation set $\mathcal{C}_p$ to which it belongs, i.e., $\mathcal{S}_k \subseteq \mathcal{C}_p$. We partition $\mathcal{P}$ in three path sets $\mathcal{P} = \mathcal{P}_{\overline{\mathcal{C}}_p} \cup \mathcal{P}_{\overline{\mathcal{S}}_k} \cup \mathcal{P}_{\mathcal{S}_k}$, as described in Equations 5.13, 5.14, and 5.15.

Proposition (ii) from Lemma 5.3 states that all path sets $\mathcal{Q} \subseteq \mathcal{P}$, for which $\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{S}}_k} \neq \emptyset$, satisfy $\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k} = 0$. Thus, the coverage indicator $\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k}$ might be different than 0 only if $\mathcal{Q} \subseteq \mathcal{P}_{\mathcal{S}_k} \cup \mathcal{P}_{\overline{\mathcal{C}}_p}$. Therefore, Equation 5.18 becomes:

$$\Gamma_{\mathcal{P},\mathcal{S}_k} = \sum_{\substack{\mathcal{Q}\subseteq\mathcal{P}_{\mathcal{S}_k}\cup\mathcal{P}_{\overline{\mathcal{C}}_p} \\ \text{s.t. } \boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k}=1}} (-1)^{|\mathcal{Q}|}. \tag{5.19}$$

Furthermore, from Lemma 5.3, Proposition (iii), we know that for any path set $\mathcal{Q} \subseteq \mathcal{P}_{\mathcal{S}_k} \cup \mathcal{P}_{\overline{\mathcal{C}}_p}$, $\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k} = 1$ if and only if $\mathcal{Q} \cap \mathcal{P}_{\mathcal{S}_k} \in \Omega_{\mathcal{P},\mathcal{S}_k}$, with $\Omega_{\mathcal{P},\mathcal{S}_k}$ given by Definition 5.0.10. Hence, we can rewrite Equation 5.19 as:

$$\Gamma_{\mathcal{P},\mathcal{S}_k} = \sum_{\substack{\mathcal{Q}\subseteq\mathcal{P}_{\mathcal{S}_k}\cup\mathcal{P}_{\overline{\mathcal{C}}_p} \\ \text{s.t. } \mathcal{Q}\cap\mathcal{P}_{\mathcal{S}_k}\in\Omega_{\mathcal{P},\mathcal{S}_k}}} (-1)^{|\mathcal{Q}|}. \tag{5.20}$$

Note that if $\Omega_{\mathcal{P},\mathcal{S}_k} = \emptyset$, then $\Gamma_{\mathcal{P},\mathcal{S}_k} = 0$, which concludes our proof. In the rest of this proof, we assume that $\Omega_{\mathcal{P},\mathcal{S}_k} \neq \emptyset$.

We group all path sets $\mathcal{Q} \subseteq \mathcal{P}_{\mathcal{S}_k} \cup \mathcal{P}_{\overline{\mathcal{C}}_p}$, with $\mathcal{Q} \cap \mathcal{P}_{\mathcal{S}_k} \in \Omega_{\mathcal{P},\mathcal{S}_k}$, by their projection on $\Omega_{\mathcal{P},\mathcal{S}_k}$, that is, Equation 5.20 reads:

$$\Gamma_{\mathcal{P},\mathcal{S}_k} = \sum_{\mathcal{O}\in\Omega_{\mathcal{P},\mathcal{S}_k}} \sum_{\substack{\mathcal{Q}\subseteq\mathcal{P}_{\mathcal{S}_k}\cup\mathcal{P}_{\overline{\mathcal{C}}_p} \\ \text{s.t. } \mathcal{Q}\cap\mathcal{P}_{\mathcal{S}_k}=\mathcal{O}}} (-1)^{|\mathcal{Q}|}. \tag{5.21}$$

For any path set $\mathcal{Q} \subseteq \mathcal{P}_{\mathcal{S}_k} \cup \mathcal{P}_{\overline{\mathcal{C}}_p}$, we can write:

$$\mathcal{Q} = (\mathcal{Q} \cap \mathcal{P}_{\mathcal{S}_k}) \cup (\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{C}}_p}) = \mathcal{O} \cup \mathcal{V},$$

where $\mathcal{O} = \mathcal{Q} \cap \mathcal{P}_{\mathcal{S}_k}$ and $\mathcal{V} = \mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{C}}_p}$. Since $\mathcal{O} \cap \mathcal{V} = \emptyset$, we can recast Equation 5.21 as:

$$\Gamma_{\mathcal{P},\mathcal{S}_k} = \sum_{\mathcal{O} \in \Omega_{\mathcal{P},\mathcal{S}_k}} \sum_{\mathcal{V} \subseteq \mathcal{P}_{\overline{\mathcal{C}}_p}} (-1)^{|\mathcal{O}|+|\mathcal{V}|} = \sum_{\mathcal{O} \in \Omega_{\mathcal{P},\mathcal{S}_k}} (-1)^{|\mathcal{O}|} \sum_{\mathcal{V} \subseteq \mathcal{P}_{\overline{\mathcal{C}}_p}} (-1)^{|\mathcal{V}|}. \quad (5.22)$$

Next, we focus on the right-most sum in Equation 5.22, and we obtain:

$$\sum_{\mathcal{V} \subseteq \mathcal{P}_{\overline{\mathcal{C}}_p}} (-1)^{|\mathcal{V}|} = \sum_{m=1}^{|\mathcal{P}_{\overline{\mathcal{C}}_p}|} \sum_{\substack{\mathcal{V} \subseteq \mathcal{P}_{\bar{\mathcal{C}}_p} \\ \text{s.t.}|\mathcal{V}|=m}} (-1)^m \quad = \sum_{m=1}^{|\mathcal{P}_{\overline{\mathcal{C}}_p}|} (-1)^m \binom{|\mathcal{P}_{\bar{\mathcal{C}}_p}|}{m} = 0, \quad (5.23)$$

because the Binomial theorem yields that $\sum_{m=1}^{n} (-1)^m \binom{n}{m} = 0$ for any positive integer $n > 0$.

Finally, by combing Equation 5.23 and Equation 5.22, we get $\Gamma_{\mathcal{P},\mathcal{S}_k} = 0$, for any potential congested correlation subset $\mathcal{S}_k$ in $\widehat{S}$, which concludes our proof.

□

Puzzle pieces

In this section, we present two lemmas which we use in the proof of Theorem 5.1.

Lemma 5.2. *For any redundant path set $\mathcal{P} \subseteq P$, and for any potentially congested correlation subset $\mathcal{S}_k \in \widehat{S}$, whose correlation set is $\mathcal{C}_p$,*

$$\mathcal{P}_{\overline{\mathcal{C}}_p} \neq \emptyset,$$

with $\mathcal{P}_{\overline{\mathcal{C}}_p}$ given by Equation 5.13.

Proof. We prove our lemma by contradiction. Assume that there is a path set $\mathcal{P} \subseteq P$, and a potentially congested correlation subset $\mathcal{S}_k \in \widehat{S}$, whose correlation set is $\mathcal{C}_p$, such that path set $\mathcal{P}$ is redundant and $\mathcal{P}_{\overline{\mathcal{C}}_p} = \emptyset$.

We define $\mathcal{S}_l = \widehat{Links}(\mathcal{P}) \cap \mathcal{C}_p$, where $\widehat{Links}(\mathcal{P})$ is the potentially congested link coverage function applied to path set $\mathcal{P}$ given by Definition 5.0.6. Note that $\mathcal{S}_l$ may be the same as or different than $\mathcal{S}_k$ since in the definition of $\mathcal{P}_{\overline{\mathcal{C}}_p}$ we only consider correlation set $\mathcal{C}_p$. Let $\bar{\mathcal{S}}_l$ be the complement of $\mathcal{S}_l$ given

by Definition 5.0.5. We will prove that path set $\mathcal{P}$ cannot be redundant because $\mathcal{S}_l \in \widehat{S}$ and $\mathcal{P} \subseteq \mathit{Paths}(\mathcal{S}_l) \setminus \mathit{Paths}(\bar{\mathcal{S}}_l)$, which contradicts Definition 5.0.9. Toward this goal, we must show that (i) $\mathcal{S}_l \in \widehat{S}$, (ii) $\mathcal{P} \subseteq \mathit{Paths}(\mathcal{S}_l)$, and (iii) $\mathcal{P} \cap \mathit{Paths}(\bar{\mathcal{S}}_l) = \emptyset$.

(i) First, we show that $\mathcal{S}_l \neq \emptyset$, and $\mathcal{S}_l \in \widehat{S}$. From the contradiction hypothesis, $\mathcal{P}_{\overline{\mathcal{C}}_p} = \emptyset$; therefore, for any path $p_i \in \mathcal{P}$, Equation 5.13 yields that:

$$\emptyset \;\neq\; \widehat{\mathit{Links}}(\{p_i\}) \cap \mathcal{C}_p \;\subseteq\; \widehat{\mathit{Links}}(\mathcal{P}) \cap \mathcal{C}_p \;=\; \mathcal{S}_l. \tag{5.24}$$

By definition, $\mathcal{S}_l$ is a correlation subset ($\mathcal{S}_l \subseteq \mathcal{C}_p$) and $\mathcal{S}_l$ contains only potentially congested links ($\mathcal{S}_l \subseteq \widehat{\mathit{Links}}(\mathcal{P})$). Since $\mathcal{S}_l \neq \emptyset$, Definition 5.0.3 yields that $\mathcal{S}_l \in \widehat{S}$.

(ii) Second, we show that $\mathcal{P} \subseteq \mathit{Paths}(\mathcal{S}_l)$. From Equation 5.24, we know that $\emptyset \;\neq\; \widehat{\mathit{Links}}(\{p_i\}) \cap \mathcal{C}_p \;\subseteq\; \mathcal{S}_l$, for all paths $p_i \in \mathcal{P}$. Hence, any path $p_i \in \mathcal{P}$ traverses at least one link in $\mathcal{S}_l$, which by the definition of the path coverage function (Definition 4.1.1), implies that $p_i \in \mathit{Paths}(\mathcal{S}_l)$.

(iii) Third, we show that $\mathcal{P} \cap \mathit{Paths}(\bar{\mathcal{S}}_l) = \emptyset$. From Definition 5.0.5:

$$\emptyset \;=\; \mathcal{S}_l \cap \bar{\mathcal{S}}_l \;=\; (\widehat{\mathit{Links}}(\mathcal{P}) \cap \mathcal{C}_p) \cap \bar{\mathcal{S}}_l = \; \widehat{\mathit{Links}}(\mathcal{P}) \cap \bar{\mathcal{S}}_l.$$

For any path $p_i \in \mathcal{P}$:

$$\widehat{\mathit{Links}}(\{p_i\}) \cap \bar{\mathcal{S}}_l \;\subseteq\; \widehat{\mathit{Links}}(\mathcal{P}) \cap \bar{\mathcal{S}}_l \;=\; \emptyset.$$

By definition, all links in $\bar{\mathcal{S}}_l$ are potentially congested. Since $\widehat{\mathit{Links}}(\{p_i\}) \cap \bar{\mathcal{S}}_l = \emptyset$, for any $p_i \in \mathcal{P}$, none of the paths in $\mathcal{P}$ traverses any link in $\bar{\mathcal{S}}_l$, which implies that $\mathcal{P} \cap \mathit{Paths}(\bar{\mathcal{S}}_l) = \emptyset$.

We conclude that there is a potentially congested correlation subset, namely $\mathcal{S}_l$, such that $\mathcal{P} \subseteq \mathit{Paths}(\mathcal{S}_l) \setminus \mathit{Paths}(\bar{\mathcal{S}}_l)$, which contradicts the fact that path set $\mathcal{P}$ is redundant. Therefore, $\mathcal{P}_{\overline{\mathcal{C}}_p} \neq \emptyset$.

□

Lemma 5.3. *Consider a path set $\mathcal{P} \subseteq P$ and a potentially congested correlation subset $\mathcal{S}_k \in \widehat{S}$, which belongs to correlation set $\mathcal{C}_p$. For any path set $\mathcal{Q} \subseteq \mathcal{P}$,*

(i) $\widehat{\mathit{Links}}(\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{C}}_p}) \cap \mathcal{C}_p = \emptyset$;
(ii) if $\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{S}}_k} \neq \emptyset$, then $\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k} = 0$;

(iii) if $\mathcal{Q} \subseteq \mathcal{P}_{\mathcal{S}_k} \cup \mathcal{P}_{\overline{\mathcal{C}}_p}$*, then* $\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k} = 1$ *if and only if* $\mathcal{Q} \cap \mathcal{P}_{\mathcal{S}_k} \in \Omega_{\mathcal{P},\mathcal{S}_k}$*;*

where $\mathcal{P} = \mathcal{P}_{\overline{\mathcal{C}}_p} \cup \mathcal{P}_{\overline{\mathcal{S}}_k} \cup \mathcal{P}_{\mathcal{S}_k}$ *is the partition of* $\mathcal{P}$ *described by Equations 5.13, 5.14, and 5.15,* $\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k}$ *is the coverage indicator given by Definition 5.0.8, and* $\Omega_{\mathcal{P},\mathcal{S}_k}$ *is given by Definition 5.0.10.*

Proof. We prove subsequently each of the three propositions.

Proposition (i). We apply the potentially congested link coverage function to path set $\mathcal{P}_{\overline{\mathcal{C}}_p}$ as given by Definition 5.0.6, and we expand the term $\widehat{Links}(\mathcal{P}_{\overline{\mathcal{C}}_p}) \cap \mathcal{C}_p$ using the rule in Equation 5.4:

$$\widehat{Links}(\mathcal{P}_{\overline{\mathcal{C}}_p}) \cap \mathcal{C}_p = \widehat{Links}(\bigcup_{p_i \in \mathcal{P}_{\overline{\mathcal{C}}_p}} p_i) \cap \mathcal{C}_p = \bigcup_{p_i \in \mathcal{P}_{\overline{\mathcal{C}}_p}} \widehat{Links}(\{p_i\}) \cap \mathcal{C}_p.$$

From Equation 5.13, $\widehat{Links}(\{p_i\}) \cap \mathcal{C}_p = \emptyset$, for all paths $p_i \in \mathcal{P}_{\overline{\mathcal{C}}_p}$. Therefore, $\widehat{Links}(\mathcal{P}_{\overline{\mathcal{C}}_p}) \cap \mathcal{C}_p = \emptyset$. Next, we apply the rule in Equation 5.3 to $\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{C}}_p} \subseteq \mathcal{P}_{\overline{\mathcal{C}}_p}$:

$$\widehat{Links}(\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{C}}_p}) \cap \mathcal{C}_p \subseteq \widehat{Links}(\mathcal{P}_{\overline{\mathcal{C}}_p}) \cap \mathcal{C}_p = \emptyset,$$

which concludes our proof of (i).

Proposition (ii). We prove this proposition by contradiction. Suppose that there is a path set $\mathcal{Q} \subseteq \mathcal{P}$, such that $\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{S}}_k} \neq \emptyset$ and $\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k} = 1$. From Definition 5.0.8, $\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k} = 1$ if and only if $\widehat{Links}(\mathcal{Q}) \cap \mathcal{C}_p = \mathcal{S}_k$. Using the rule in Equation 5.3 we obtain:

$$\mathcal{S}_k = \widehat{Links}(\mathcal{Q}) \cap \mathcal{C}_p \supseteq \widehat{Links}(\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{S}}_k}) \cap \mathcal{C}_p,$$

where $\overline{\mathcal{S}}_k$ is the complement of $\mathcal{S}_k$ given by Definition 5.0.5. Since $\overline{\mathcal{S}}_k \subseteq \mathcal{C}_p$, the above expression implies that:

$$\mathcal{S}_k \supseteq \widehat{Links}(\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{S}}_k}) \cap \overline{\mathcal{S}}_k.$$

On the other hand, $\mathcal{S}_k \cap \overline{\mathcal{S}}_k = \emptyset$; hence, the only possible explanation is that $\widehat{Links}(\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{S}}_k}) \cap \overline{\mathcal{S}}_k = \emptyset$. This implies that either $\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{S}}_k} = \emptyset$ or $\overline{\mathcal{S}}_k = \emptyset$. However, if $\overline{\mathcal{S}}_k = \emptyset$, then by Equations 5.14, $\mathcal{P}_{\overline{\mathcal{S}}_k} = \emptyset$, and consequently, $\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{S}}_k} = \emptyset$. Therefore, $\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{S}}_k} = \emptyset$, which contradicts our hypothesis.

Proposition (iii). Since we know from the hypothesis that $\mathcal{Q} \subseteq \mathcal{P}_{\mathcal{S}_k} \cup \mathcal{P}_{\overline{\mathcal{C}}_p}$, using the rules in Equations 5.4 and 5.3, we can expand the term $\widehat{Links}(\mathcal{Q}) \cap \mathcal{C}_p$ as:

$$\begin{aligned}
\widehat{Links}(\mathcal{Q}) \cap \mathcal{C}_p &= \widehat{Links}(\mathcal{Q} \cap (\mathcal{P}_{\overline{\mathcal{C}}_p} \cup \mathcal{P}_{\mathcal{S}_k})) \cap \mathcal{C}_p \\
&= \widehat{Links}((\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{C}}_p}) \cup (\mathcal{Q} \cap \mathcal{P}_{\mathcal{S}_k})) \cap \mathcal{C}_p \\
&= \Big(\widehat{Links}(\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{C}}_p}) \cup \widehat{Links}(\mathcal{Q} \cap \mathcal{P}_{\mathcal{S}_k})\Big) \cap \mathcal{C}_p \\
&= \big(\widehat{Links}(\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{C}}_p}) \cap \mathcal{C}_p\big) \cup \big(\widehat{Links}(\mathcal{Q} \cap \mathcal{P}_{\mathcal{S}_k}) \cap \mathcal{C}_p\big).
\end{aligned}$$

From Proposition (i), $\widehat{Links}(\mathcal{Q} \cap \mathcal{P}_{\overline{\mathcal{C}}_p}) \cap \mathcal{C}_p = \emptyset$; thus:

$$\widehat{Links}(\mathcal{Q}) \cap \mathcal{C}_p = \widehat{Links}(\mathcal{Q} \cap \mathcal{P}_{\mathcal{S}_k}) \cap \mathcal{C}_p. \tag{5.25}$$

By Definition 5.0.8 and Equation 5.25, we conclude that:

$$\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k} = \boldsymbol{\alpha}_{\mathcal{Q} \cap \mathcal{P}_{\mathcal{S}_k},\mathcal{S}_k} \tag{5.26}$$

Because of Equation 5.26, if $\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k} = 1$, then $\boldsymbol{\alpha}_{\mathcal{Q} \cap \mathcal{P}_{\mathcal{S}_k},\mathcal{S}_k} = \boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k} = 1$; hence, $\mathcal{Q} \cap \mathcal{P}_{\mathcal{S}_k} \in \Omega_{\mathcal{P},\mathcal{S}_k}$ by Definition 5.0.10. Conversely, if $\mathcal{Q} \cap \mathcal{P}_{\mathcal{S}_k} \in \Omega_{\mathcal{P},\mathcal{S}_k}$, then Definition 5.0.10 implies that $\boldsymbol{\alpha}_{\mathcal{Q} \cap \mathcal{P}_{\mathcal{S}_k},\mathcal{S}_k} = 1$, and Equation 5.26 yields $\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k} = 1$. Hence, $\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k} = 1$ if and only if $\mathcal{Q} \cap \mathcal{P}_{\mathcal{S}_k} \in \Omega_{\mathcal{P},\mathcal{S}_k}$. □

5.2.4 The Algorithm

In this section, based on the theoretical results presented in Section 5.2.3, we propose an algorithm that solves Congestion Probability Inference. Our algorithm finds the maximum number of linearly independent equations possible, by applying Equation 5.6 to "wisely chosen" non-redundant sets of paths (Definition 5.0.9).

Our algorithm computes the probability that all links in $\mathcal{S}_k$ are good, for all potentially congested correlation subsets $\mathcal{S}_k \in \widehat{S}$, with $\widehat{S}$ given by Definition 5.0.4. If we know these probabilities, we can easily determine the congestion probabilities of all sets of links. For example, in Figure 5.1, Scenario B, if

we know the probabilities $\mathbb{P}\left(Z_{e_j}=0\right)$ with $j=1\ldots4$, and $\mathbb{P}\left(Z_{e_3}=0,Z_{e_4}=0\right)$, we can compute:

$$\mathbb{P}\left(Z_{e_j}=1\right)=1-\mathbb{P}\left(Z_{e_j}=0\right),\ j=1\ldots4,$$
$$\mathbb{P}\left(Z_{e_3}=1,Z_{e_4}=1\right)=1-\mathbb{P}\left(Z_{e_3}=1\right)-\mathbb{P}\left(Z_{e_4}=1\right)+\mathbb{P}\left(Z_{e_3}=0,Z_{e_4}=0\right).$$

Since we assume that links which belong to different correlation sets are independent, the congestion probability of the other sets of links can be expressed as products of these 5 probabilities, i.e.,

$$\mathbb{P}\left(Z_{e_1}=1,Z_{e_2}=1\right)=\mathbb{P}\left(Z_{e_1}=1\right)\mathbb{P}\left(Z_{e_2}=1\right).$$

The input to the algorithm is any ordering $\widehat{S}$ of all the potentially congested correlation subsets. The output is an ordering of path sets $\widehat{\mathcal{P}}$ to which we apply Equation 5.6 to form a system as depicted in Equation 5.8. If the Identifiability++ condition holds, then this system has an unique solution, hence, we can solve it and determine the quantities of interest.

First, we form an initial list of path sets $\widehat{\mathcal{P}}$ (lines 1 to 4). We ensure that each correlation subset $\mathcal{S}_k \in \widehat{S}$ is traversed by at least one of the path sets in $\widehat{\mathcal{P}}$, namely, path set $Paths(\mathcal{S}_k) \setminus Paths(\overline{\mathcal{S}}_k)$, where $\overline{\mathcal{S}}_k$ is the complement of $\mathcal{S}_k$ given by Definition 5.0.5 (lines 2 and 3). We illustrate with an example. Suppose that, in Figure 5.1, all correlation subsets are potentially congested and we pick ordering $\widehat{S}=\langle\{e_1\},\{e_2\},\{e_3\},\{e_4\},\{e_3,e_4\}\rangle$. After line 4 has been executed, $\widehat{\mathcal{P}}$ consists of the path sets in the last column of the following table:

$\mathcal{S}_k$	$\overline{\mathcal{S}}_k$	$Paths(\mathcal{S}_k)$	$Paths(\overline{\mathcal{S}}_k)$	$Paths(\mathcal{S}_k) \setminus Paths(\overline{\mathcal{S}}_k)$
$\{e_1\}$	$\emptyset$	$\{p_1,p_2\}$	$\emptyset$	$\{p_1,p_2\}$
$\{e_2\}$	$\emptyset$	$\{p_3\}$	$\emptyset$	$\{p_3\}$
$\{e_3\}$	$\{e_4\}$	$\{p_1\}$	$\{p_2,p_3\}$	$\{p_1\}$
$\{e_4\}$	$\{e_3\}$	$\{p_2,p_3\}$	$\{p_1\}$	$\{p_2,p_3\}$
$\{e_3,e_4\}$	$\emptyset$	$\{p_1,p_2,p_3\}$	$\emptyset$	$\{p_1,p_2,p_3\}$

Algorithm 5.1 *Selection of Path Sets*

Input: $\widehat{S}$: a list of potentially congested correlation subsets
Variables: $\widehat{\mathcal{P}}$: a list of path sets
$\mathcal{P}$: a path set
$\mathcal{S}_k$: a correlation subset

1: $\widehat{\mathcal{P}} \leftarrow \langle\rangle$
2: **for all** $\mathcal{S}_k \in \widehat{S}$ **do**
3: $\quad \mathcal{P} \leftarrow \mathit{Paths}(\mathcal{S}_k) \setminus \mathit{Paths}(\overline{\mathcal{S}}_k)$
4: $\quad \widehat{\mathcal{P}} \leftarrow \widehat{\mathcal{P}} + \mathcal{P}$

5: $\mathbf{A} \leftarrow \mathit{Matrix}(\widehat{\mathcal{P}}, \widehat{S})$
6: $\mathbf{N} \leftarrow \mathit{NullSpace}\,(\mathbf{A})$

7: **repeat**
8: $\quad$ **for all** $\mathcal{S}_k \in \mathit{SortByHammingWeight}(\widehat{S}, \mathbf{N})$ **do**
9: $\quad\quad$ **for all** $\mathcal{P} \subseteq \mathit{Paths}(\mathcal{S}_k) \setminus \mathit{Paths}(\overline{\mathcal{S}}_k)$ **do**
10: $\quad\quad\quad \mathbf{r} \leftarrow \mathit{Row}(\mathcal{P}, \widehat{S})$
11: $\quad\quad\quad$ **if** $||\mathbf{rN}|| > 0$ **then**
12: $\quad\quad\quad\quad \widehat{\mathcal{P}} \leftarrow \widehat{\mathcal{P}} + \mathcal{P}$
13: $\quad\quad\quad\quad \mathbf{N} \leftarrow \mathit{NullSpaceUpdate}\,(\mathbf{N}, \mathbf{r})$
14: $\quad\quad\quad\quad$ go to line 15
15: **until** $\mathbf{N}$ has no columns left

16: **return** $\widehat{\mathcal{P}}$

Notation:
$\mathcal{A} \setminus \mathcal{B}$: subtract set $\mathcal{B}$ from set $\mathcal{A}$
$\widehat{\mathcal{P}} + \mathcal{P}$: add path set $\mathcal{P}$ to list of path sets $\widehat{\mathcal{P}}$

That is, $\widehat{\mathcal{P}} = \{\{p_1, p_2\}, \{p_3\}, \{p_1\}, \{p_2, p_3\}, \{p_1, p_2, p_3\}\}$. If we apply Equation 5.6 to each of the path sets in $\widehat{\mathcal{P}}$, we obtain:

$$\mathbb{P}\,(W_{p_1} = 0, W_{p_2} = 0) = \mathbb{P}\,(Z_{e_1} = 0)\,\mathbb{P}\,(Z_{e_3} = 0, Z_{e_4} = 0)$$
$$\mathbb{P}\,(W_{p_3} = 0) = \mathbb{P}\,(Z_{e_2} = 0)\,\mathbb{P}\,(Z_{e_4} = 0)$$
$$\mathbb{P}\,(W_{p_1} = 0) = \mathbb{P}\,(Z_{e_1} = 0)\,\mathbb{P}\,(Z_{e_3} = 0)$$
$$\mathbb{P}\,(W_{p_2} = 0, W_{p_3} = 0) = \mathbb{P}\,(Z_{e_1} = 0)\,\mathbb{P}\,(Z_{e_2} = 0)\,\mathbb{P}\,(Z_{e_4} = 0)$$
$$\mathbb{P}\,(W_{p_1} = 0, W_{p_2} = 0, W_{p_3} = 0) = \mathbb{P}\,(Z_{e_1} = 0)\,\mathbb{P}\,(Z_{e_2} = 0)\,\mathbb{P}\,(Z_{e_3} = 0, Z_{e_4} = 0)$$

Next, we take the logarithm of each of these equation as depicted in Equation 5.7 and we derive the system in Equation 5.8:

$$\underbrace{\begin{bmatrix} \log\mathbb{P}\left(W_{p_1}=0, W_{p_2}=0\right) \\ \log\mathbb{P}\left(W_{p_3}=0\right) \\ \log\mathbb{P}\left(W_{p_1}=0\right) \\ \log\mathbb{P}\left(W_{p_2}=0, W_{p_3}=0\right) \\ \log\mathbb{P}\left(W_{p_1}=0, W_{p_2}=0, W_{p_3}=0\right) \end{bmatrix}}_{\mathbf{V}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}}_{Matrix(\widehat{\mathcal{P}},\widehat{S})} \underbrace{\begin{bmatrix} \log\mathbb{P}\left(Z_{e_1}=0\right) \\ \log\mathbb{P}\left(Z_{e_2}=0\right) \\ \log\mathbb{P}\left(Z_{e_3}=0\right) \\ \log\mathbb{P}\left(Z_{e_4}=0\right) \\ \log\mathbb{P}\left(Z_{e_3}=0, Z_{e_4}=0\right) \end{bmatrix}}_{\mathbf{U}}.$$

In this case, $Matrix(\widehat{\mathcal{P}}, \widehat{S})$ in Equation 5.8 has full column rank, which means that we can solve our system and compute, for each correlation subset, the probability that all links in that subset are good, hence, also the congestion probability of each set of links in the network. In general, however, the resulting system of equations is under-determined, and we continue with the second part of the algorithm.

We augment the initial list of path sets $\widehat{\mathcal{P}}$ by iteratively adding path sets such that we increase the rank of the associated matrix $Matrix(\widehat{\mathcal{P}}, \widehat{S})$ (lines 5 to 15). More specifically, we first compute the matrix $\mathbf{A}$ associated with the initial list of path sets $\widehat{\mathcal{P}}$ (line 5), as well as a matrix $\mathbf{N}$, whose columns span the null space of $\mathbf{A}$ (line 6); the latter can be done using standard techniques, like singular value decomposition or QR factorization. Next, we iteratively identify a path set $\mathcal{P}$ such that adding $\mathbf{r} = Row(\mathcal{P}, \widehat{S})$ (Definition 5.0.7) to the system matrix, increases the latter's rank, and we add $\mathcal{P}$ to $\widehat{\mathcal{P}}$ (lines 10 to 12). Every time we add a new path set to $\widehat{\mathcal{P}}$, we update the matrix $\mathbf{N}$, such that its columns always span the null space of $Matrix(\widehat{\mathcal{P}}, \widehat{S})$ (line 13). We stop the iteration when $\mathbf{N}$ is left with 0 columns, i.e., the loop in line 7 finishes (line 15).

Selecting a Useful Path Set

We first discuss the problem of identifying a new set of paths $\mathcal{P}$ such that $Row(\mathcal{P}, \widehat{S})$ (Definition 5.0.7) increases the rank of the system matrix.

The fundamental theorem of linear algebra states that the null space of a matrix is the orthogonal complement of its row space. In our case, in order for a vector $\mathbf{r}$ to increase the rank of the system matrix, it is necessary and sufficient that $\mathbf{r}$ is not orthogonal to the null space basis $\mathbf{N}$, i.e., $||\mathbf{rN}|| > 0$. Therefore, we are looking for a path set $\mathcal{P}$ such that the vector $Row(\mathcal{P}, \widehat{S})$ satisfies this condition. If such a path set exists, our algorithm is guaranteed to find it, because it iterates over all non-redundant (Definition 5.0.9) sets of paths (lines

8 and 9) and tests whether each of them satisfies the corresponding condition (lines 10 and 11).

However, to save time, the algorithm orders the sets of paths such that it first tries those that are more *likely* to satisfy the condition (this is the role of the *SortByHammingWeight* function). Intuitively, if the i-th element of vector $\mathbf{r}$ is non-zero and the i-th row of matrix $\mathbf{N}$ has many non-zero elements, then $||\mathbf{rN}|| > 0$ is likely to be true. Thus, our algorithm picks the row of $\mathbf{N}$ with the largest number of non-zero elements (the largest Hamming weight); suppose that this row corresponds to correlation subset $\mathcal{S}_k$ (line 8). Then, it looks for any path set $\mathcal{P}$ which traverses $\mathcal{S}_k$ (line 9), and picks the first one that satisfies the condition (lines 10 to 11). The *SortByHammingWeight* helps us pick the correlation subset $\mathcal{S}_k$—it outputs an ordering of the correlation subsets in $\widehat{S}$ such that the first element in that ordering corresponds to the row of matrix $\mathbf{N}$ with the largest Hamming weight. Therefore, so far we have considered any potentially ordering of the correlation subsets $\widehat{S}$; now we choose this particular ordering in order to speed up the computation.

Nevertheless, iterating over all possible subsets of path set $Paths(\mathcal{S}_k) \setminus Paths(\overline{\mathcal{S}}_k)$, for all $\mathcal{S}_k \in \widehat{S}$, can be time consuming, therefore, in order to speed up the computation, we propose the heuristic described in Section A.1. Even though this heuristic does not consider all possible path sets, in practice, it always find the maximum number of linearly independent rows. Furthermore, this heuristic can be used in combination with Algorithm 5.1: if the heuristic cannot find all linearly independent rows, we can always switch back to the complete version of the algorithm to find the missing rows, if any, without loosing the path sets already discovered by the heuristic.

Updating the Null Space Basis

In this section, we present an algorithm which incrementally updates the null space basis of the system's matrix. Computing the null space of a matrix with thousands of rows takes a significant amount of time, and doing this at every iteration would render the algorithm practically useless. Instead, the $NullSpaceUpdate$ function (Algorithm 5.2) updates the null space incrementally, i.e., given the null space computed in the previous iteration, it efficiently updates the null space. The correctness of this algorithm is ensured by Lemma 5.4.

Algorithm 5.2 *NullSpaceUpdate*

Input: $\mathbf{N}$: a matrix of size $n \times m$
$\mathbf{r}$: a row vector of n elements

1: **return** $\left(\mathbf{I}_n - \frac{\mathbf{N}_{*1}\mathbf{r}}{\mathbf{r}\mathbf{N}_{*1}}\right)\mathbf{N}_{*2:m}$

Notation:
$\mathbf{I}_n$: the identity matrix of size n
$\mathbf{N}_{*1}$: the 1-st column of matrix $\mathbf{N}$
$\mathbf{N}_{*2:m}$: the matrix formed by taking columns 2 to m of $\mathbf{N}$

Lemma 5.4. *A matrix* $\mathbf{A}$ *of dimension* $m \times n$ *and rank* u*, is expanded with a row* $\mathbf{r}$ *that increases the rank of the matrix, i.e.,*

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A} \\ \mathbf{r} \end{bmatrix}, \tag{5.27}$$

where $rank(\mathbf{A}') = u + 1$*. Then, a basis*[1] *of the null space of* $\mathbf{A}'$ *can be computed as:*

$$\mathbf{N}' = \left(\mathbf{I}_n - \frac{\mathbf{N}_{*1}\mathbf{r}}{\mathbf{r}\mathbf{N}_{*1}}\right)\mathbf{N}_{*2:(n-u)}, \tag{5.28}$$

where $\mathbf{I}_n$ *is the identity matrix of dimension* n*,* $\mathbf{N}$ *is a basis of the null space of matrix* $\mathbf{A}$*,* $\mathbf{N}_{*1}$ *is the first column of matrix* $\mathbf{N}$*, and* $\mathbf{N}_{*2:(n-u)}$ *is the matrix formed by columns* 2 *to* $(n-u)$ *of matrix* $\mathbf{N}$*.*

Proof. The null space of matrix $\mathbf{A}$ has dimension $n-u$, while the null space of matrix $\mathbf{A}'$ has dimension $n-u-1$. We denote by $\mathbf{N}$ a basis of the null space of $\mathbf{A}$, and by $\mathbf{N}'$ a basis of the null space of $\mathbf{A}'$. The matrix $\mathbf{N}'$ must satisfy two conditions: (i) each column of $\mathbf{N}'$ is orthogonal to each row of $\mathbf{A}'$ (the null space of a matrix is the orthogonal complement of its row space), and (ii) the rank of $\mathbf{N}'$ must be $n-u-1$.

The null space of matrix $\mathbf{A}'$ is a subspace of the null space of $\mathbf{A}$. Therefore, we can write each column of $\mathbf{N}'$ as a linear combination of the columns of $\mathbf{N}$:

$$\mathbf{N}' = \mathbf{N}\mathbf{T}, \tag{5.29}$$

where $\mathbf{T}$ is a transformation matrix of dimension $(n-u) \times (n-u-1)$. Equation 5.29 ensures that each column of $\mathbf{N}'$ is orthogonal to each row of $\mathbf{A}$, but for

[1] We write a basis as a matrix whose columns are the vectors of the basis.

Condition (i) to hold, we also require that the new row vector $\mathbf{r}$ is orthogonal to $\mathbf{N}'$:

$$\mathbf{r}\mathbf{N}' = \mathbf{0}. \tag{5.30}$$

We combine Equations 5.29 and 5.30, and we obtain:

$$\mathbf{r}\mathbf{N}\mathbf{T} = \mathbf{0}. \tag{5.31}$$

Since we assume that we know a basis $\mathbf{N}$ of the null space of matrix $\mathbf{A}$ and the row vector $\mathbf{r}$, we can determine $\mathbf{r}\mathbf{N}$. In fact, Equation 5.31 is an undetermined system of linear equations with $n-u$ equations, and $(n-u)\cdot(n-u-1)$ unknowns, i.e., the entries in the transformation matrix $\mathbf{T}$.

We consider a transformation matrix $\mathbf{T}$ of the following form:

$$\mathbf{T} = \begin{bmatrix} \beta \\ \mathbf{I}_{n-u-1} \end{bmatrix}, \tag{5.32}$$

where $\mathbf{I}_{n-u-1}$ is the identity matrix of dimension $n-u-1$, and β a row matrix of dimension $1 \times (n-u-1)$. We choose this form for the transformation matrix $\mathbf{T}$ to ensure that the rank of $\mathbf{T}$ is $n-u-1$, and consequently, that $\mathbf{N}'$ has full column rank, as required by Condition (ii).

Next, we use the transformation matrix defined in Equation 5.32, to solve the system of linear equations in Equation 5.31. After some algebraic manipulations, we obtain:

$$\beta = -\frac{\mathbf{r}\mathbf{N}_{*2:(n-u)}}{\mathbf{r}\mathbf{N}_{*1}}, \tag{5.33}$$

where $\mathbf{N}_{*1}$ is the first column of matrix $\mathbf{N}$, and $\mathbf{N}_{*2:(n-u)}$ is the matrix formed by columns 2 to $(n-u)$ of matrix $\mathbf{N}$.

In conclusion, we have determined a transformation matrix $\mathbf{T}$, which we can use in Equation 5.29 to find a basis $\mathbf{N}'$ of the null space of $\mathbf{A}'$. The formula that we obtain is the one in Equation 5.28. □

Properties of the Algorithm

In this section, we discuss a few important properties of our algorithm.

Completeness. Our algorithm is complete in the sense that if we apply Equation 5.7 to all path sets in the ordering $\widehat{\mathcal{P}}$ returned by the algorithm, we obtain the maximum number of linearly independent equations (which can be

obtained by applying Equation 5.7 to all sets of paths). This is true regardless whether the Identifiability++ condition holds or not. The proof is immediate: we have seen in Section 5.2 (Theorem 5.1) that Equation 5.7 applied to a redundant path set $\mathcal{P}$ is a linear combinations of Equation 5.7 applied to all non-redundant path sets. Furthermore, by definition, a redundant path set is one for which there is no potentially congested correlation subset $\mathcal{S}_k \in \widehat{S}$ such that $\mathcal{P} \subseteq \mathit{Paths}(\mathcal{S}_k) \setminus \mathit{Paths}(\overline{\mathcal{S}}_k)$. Our algorithm iterates over all potentially congested correlation subsets $\mathcal{S}_k \in \widehat{S}$ (line 8 of Algorithm 5.1), and over all sets of paths included in $\mathit{Paths}(\mathcal{S}_k) \setminus \mathit{Paths}(\overline{\mathcal{S}}_k)$ (line 9 of Algorithm 5.1), therefore it covers all path sets that are not redundant, i.e., all path sets which might generate linearly independent equations.

Complexity. The complexity of our algorithm is $\mathcal{O}(n_1^3 + n_1^2 2^{n_2} n_3)$, where $n_1 = |\widehat{S}|$ is the number of potentially congested correlation subsets, $n_2 = \max_{\mathcal{S}_k \in \widehat{S}} |\mathit{Paths}(\mathcal{S}_k)|$ is the maximum number of paths which traverse the same potentially congested correlation subset (see Definition 4.1.1), n_3 is the nullity of the initial system matrix $\mathbf{A}$. We express complexity as a function of these three parameters, because any one of them can dominate the other two, depending on the topology and the congestion scenario.

The first term, $\mathcal{O}(n_1^3)$, is the complexity of the initial computation of the null space (line 5) using singular value decomposition. This is a standard operation, and we need to do it only once, in the beginning of the algorithm.

The second term, $\mathcal{O}(n_1^2 2^{n_2} n_3)$, is the complexity of searching for linearly independent rows (lines 7 to 15): We iterate over all n_1 potentially congested correlation subsets, and for each such correlation subset, we consider all possible sets of paths that cover it, which is bounded by 2^{n_2}. Furthermore, for each considered path set, we test for the condition in line 11, which has complexity $\mathcal{O}(n_1 n_3)$.

We have experimental evidence that a tighter upper bound exists: The second step, i.e., searching for linearly independent rows, can be replaced by the heuristic described in Section A.1, which is always able to find all linearly independent rows needed, and has complexity $\mathcal{O}(n_4 n_1^3 n_3)$, where $n_4 = \max_{\mathcal{S}_k \in \widehat{S}} |\mathcal{S}_k|$ is the maximum size of a potentially congested correlation subset.

When the Identifiability++ condition does not hold. Our algorithm is independent of the Identifiability++ condition in the sense that even if the Identifiability++ condition does not hold, if we apply Equation 5.6 to all path sets in the ordering $\widehat{\mathcal{P}}$ returned by the algorithm, we obtain the maximum number of linearly independent equations. The difference is that if the Identifiability++

condition holds, then we obtain a system of equations with an unique solution, which ensures that we can compute the congestion probabilities of all sets of links. If the Identifiability++ condition does not hold, then we might not be able to compute all these probabilities as stated by Theorem 4.1.

The main reason why we cannot completely solve Congestion Probability Inference when the Identifiability++ condition does not hold, is that the resulting system of equations might be undetermined. We have already discussed such an example for the toy topology in Figure 4.6 in Section 4.3. In this case, our algorithm terminates when there are no more path sets to consider, i.e., the loop in line 8 finishes without finding any new path set. This condition replaces the condition in line 15 of Algorithm 5.1.

Another reason why we cannot completely solve Congestion Probability Inference when the Identifiability++ condition does not hold, is that some correlation subsets do not appear if we apply Equation 5.6 to all possible sets of paths (see Lemma 5.5). In this case, the resulting system of equations may or may not be under-determined. Consider the toy topology in Figure 5.2, where the Identifiability++ condition does not hold as correlation subsets $\{e_1, e_2\}$ and $\{e_2\}$ are traversed by the same paths, namely, $\{p_1, p_2\}$. Suppose that all links are potentially congested, then a possible ordering of the potentially congested correlation subsets is $\widehat{S} = \{\{e_1\}, \{e_2\}, \{e_1, e_2\}, \{e_3\}\}$. If we apply Equation 5.6 to all possible set of paths, we obtain:

$$\begin{aligned}
\mathbb{P}\left(W_{p_1} = 0\right) &= \mathbb{P}\left(Z_{e_1} = 0, Z_{e_2} = 0\right) \\
\mathbb{P}\left(W_{p_2} = 0\right) &= \mathbb{P}\left(Z_{e_2} = 0\right)\mathbb{P}\left(Z_{e_3} = 0\right) \\
\mathbb{P}\left(W_{p_1} = 0, W_{p_2} = 0\right) &= \mathbb{P}\left(Z_{e_1} = 0, Z_{e_2} = 0\right)\mathbb{P}\left(Z_{e_3} = 0\right).
\end{aligned}$$

Since the probability that link e_1 is good, i.e., $\mathbb{P}\left(Z_{e_1} = 0\right)$, does not appear in any of these equations, we cannot estimate its value. However, the system of equations has 3 unknowns and 3 linearly independent equations, thus, we can compute all probabilities that appear in these equations. Therefore, we can determine the congestion probabilities $\mathbb{P}\left(Z_{e_2} = 1\right)$ and $\mathbb{P}\left(Z_{e_3} = 1\right)$, but not the probabilities $\mathbb{P}\left(Z_{e_1} = 1\right)$ and $\mathbb{P}\left(Z_{e_1} = 1, Z_{e_2} = 1\right)$. As stated by Lemma 5.5, the reason why link e_1 does not appear in any equation is because there is no sets of paths which covers this link, without covering other links from the same correlation set, i.e.,

$$\mathit{Paths}(\{e_1\}) \setminus \mathit{Paths}(\{\overline{e_1}\}) = \mathit{Paths}(\{e_1\}) \setminus \mathit{Paths}(\{e_2\}) = \{p_1\} \setminus \{p_1, p_2\} = \emptyset.$$

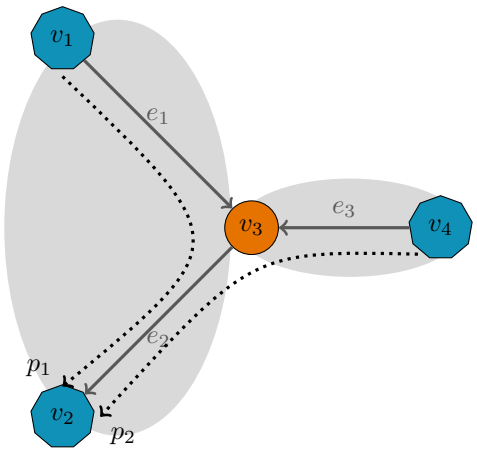


Figure 5.2: A toy topology with correlated links where the Identifiability++ condition does not hold, i.e., correlation subsets $\{e_2\}$ and $\{e_1, e_2\}$ are traversed by the same paths $\{p_1, p_2\}$. Hosts $V^H = \{v_1, v_2, v_4\}$. Routers $V^R = \{v_3\}$. Links $E = \{e_1, e_2, e_3\}$. Paths $P = \{p_1, p_2\}$. Correlation sets $C = \{\{e_1, e_2\}, \{e_3\}\}$. Correlation subsets $S = \{\{e_1\}, \{e_2\}, \{e_1, e_2\}, \{e_3\}\}$.

Therefore, when the Identifiability++ condition does not hold, we remove from the list of potentially congested correlation subsets $\widehat{S}$, all correlation subsets $\mathcal{S}_k$ for which $\mathit{Paths}(\mathcal{S}_k) \setminus \mathit{Paths}(\overline{\mathcal{S}}_k) = \emptyset$, since we know from Lemma 5.5 that these correlation subsets cannot appear in any equation. This will be the new input to Algorithm 5.1. In the example in Figure 5.2, a possible input to our algorithm is $\widehat{S} = \{\{e_2\}, \{e_1, e_2\}, \{e_3\}\}$.

Nevertheless, as we will see in Section 5.2.5, even if the Identifiability++ condition does not hold, we can still compute accurately the congestion probability of most of the potentially congested correlation subsets $\mathcal{S}_k \in \widehat{S}$ for which the Identifiability++ condition holds, that is, when there is no other correlation subset $\mathcal{S}_l \in \widehat{S}$ such that $\mathit{Paths}(\mathcal{S}_k) = \mathit{Paths}(\mathcal{S}_l)$. Therefore, whether the Identifiability++ condition holds or not for a correlation subset $\mathcal{S}_k$ is an indication of whether the congestion probability of $\mathcal{S}_k$ can be likely estimated accurately or not.

In order to apply our algorithm to a network where the Identifiability++ condition does not hold, we need to make the following small changes to Algorithm 5.1:

▷ remove from $\widehat{S}$ all correlation subsets $\mathcal{S}_k$ for which $\mathit{Paths}(\mathcal{S}_k) \setminus \mathit{Paths}(\overline{\mathcal{S}}_k) = \emptyset$; this is the new input of the algorithm.

▷ replace the condition in line 15, by the condition that the loop in line 8 finishes without finding any new path set.

Lemma 5.5. *A correlation subset $\mathcal{S}_k \in \widehat{S}$, for which $Paths(\mathcal{S}_k) \setminus Paths(\overline{\mathcal{S}}_k) = \emptyset$, satisfies $\boldsymbol{\alpha}_{\mathcal{P},\mathcal{S}_k} = 0$, for all path sets $\mathcal{P} \subseteq P$, with $\boldsymbol{\alpha}_{\mathcal{P},\mathcal{S}_k}$ given by Definition 5.0.8.*

Proof. Consider a correlation subset $\mathcal{S}_k \in \widehat{S}$, and a path set $\mathcal{P} \subseteq P$. Condition $Paths(\mathcal{S}_k) \setminus Paths(\overline{\mathcal{S}}_k) = \emptyset$ implies that $Paths(\mathcal{S}_k) \subseteq Paths(\overline{\mathcal{S}}_k)$, that is, if a path traverses a link in $\mathcal{S}_k$, then it also traverses a link in $\overline{\mathcal{S}}_k$. We consider any path set $\mathcal{P} \subseteq P$, and we partition it as described in Equations 5.13, 5.14, and 5.15. We obtain that $\mathcal{P}_{\mathcal{S}_k} = \emptyset$ since there is no path which traverses a link in $\mathcal{S}_k$, without traversing a link in $\overline{\mathcal{S}}_k$. Consequently, Definition 5.0.10 yields $\Omega_{\mathcal{P},\mathcal{S}_k} = \emptyset$. From Lemma 5.3, Propositions (ii) and (iii), we know that if $\Omega_{\mathcal{P},\mathcal{S}_k} = \emptyset$, then $\boldsymbol{\alpha}_{\mathcal{P},\mathcal{S}_k} = 0$.

In conclusion, if $Paths(\mathcal{S}_k) \setminus Paths(\overline{\mathcal{S}}_k) = \emptyset$, then $\boldsymbol{\alpha}_{\mathcal{P},\mathcal{S}_k} = 0$ for all path sets $\mathcal{P} \subseteq P$. Hence, correlation subset $\mathcal{S}_k \in \widehat{S}$ does not appear in Equation 5.6 applied to all possible sets of paths. □

When correlation sets are too large. The main challenge of our algorithm is the size of the correlation sets. If a correlation set is too large, then it is not possible to compute the congestion probability of *all* potentially congested correlation subsets. For example, given a correlation set $\mathcal{C}_p$, if all links in $\mathcal{C}_p$ are potentially congested, then we need to compute $2^{|\mathcal{C}_p|}$ probabilities, where $|\mathcal{C}_p|$ is the number of links in correlation set $\mathcal{C}_p$.

We deal with this challenge by computing the congestion probability of only *some* of the potentially congested correlation subsets. For example, we can configure our algorithm to compute only the congestion probability of individual links, i.e., the congestion probability of the correlation subsets $\mathcal{S}_k \in S$, for which $|\mathcal{S}_k| = 1$. Similarly, we can configure our algorithm to compute only the congestion probability of individual links and of pairs of links, i.e., the congestion probability of the correlation subsets $\mathcal{S}_k \in S$, for which $|\mathcal{S}_k| \leq 2$. To a certain extent, we are free to choose the correlation subsets for which we want to compute the congestion probabilities. Suppose we are interested to compute the congestion probabilities of only some particular correlation subsets. In this case, we must do the following changes to Algorithm 5.1:

▷ remove from $\widehat{S}$ the correlation subsets we are not interested in; this is the new input to the algorithm.

▷ a path set $\mathcal{P}$ is added to $\widehat{\mathcal{P}}$ (lines 4 and 12) if and only if $\boldsymbol{\alpha}_{\mathcal{P},\mathcal{S}_k} = 0$ for all correlation subsets that we have removed from $\widehat{S}$, i.e., none of the removed correlation subsets appears in Equation 5.6 applied to any path set $\mathcal{P} \in \widehat{\mathcal{P}}$.

However, in this case, even if the Identifiability++ condition holds, depending on the set of correlation subsets we are interested in, the resulting system of equations might not have a unique solution as we discard some of the path sets. This defines the extent of freedom we have to choose the correlation subsets we are interested in. On the other hand, for more "conservative" choices of the correlation subsets we are interested in, e.g., we want to compute the congestion probability of individual links, we have practical evidence that when the Identifiability++ condition holds, we can indeed compute all these probabilities.

5.2.5 Evaluation

We now look at the performance of our algorithm and compare it to the tomographic algorithm in [NT07a], which computes the congestion probability of individual links under the assumption that all links are independent. In order to better distinguish between the two, we label our algorithm *Correlation* because it assumes Correlation Sets, and the algorithm in [NT07a] *Independence* because it assumes Link Independence.

As explained in Section 5.2.4, we can configure our algorithm to compute the congestion probabilities of only some of the correlation subsets. Depending on the scenario we are simulating, we configure our algorithm to infer only the congestion probability of individual links similar to the Independence algorithm, or the congestion probability of all correlation subsets, or when the correlation sets are too large, the congestion probabilities of a maximum number of correlation subsets depending on our computational resources[2].

Metrics. To evaluate the performance of each algorithm, we look at the absolute error between the actual congestion probability of a link or of a correlation subset and the congestion probability as computed by the algorithm. For instance, if the actual congestion probability of a correlation subset is 0.5, but the algorithm thinks it is 0.1, then the absolute error is 0.4.

We use two ways to illustrate the performance of each algorithm: (i) We plot the mean of the absolute error for all potentially congested links (Definitions 5.0.2) or potentially congested correlation subsets (Definition 5.0.3). (ii) We plot the cumulative distribution function (CDF) of the absolute error,

[2]Currently, on an Intel Core Quad @ 2.4GHz machine, our algorithm can compute roughly the congestion probabilities of 5000 correlation subsets

for all the potentially congested links/ correlation subsets. For a perfect algorithm, this CDF would be a single point at $x = 0$, $y = 100\%$. In general, the earlier the CDF hits the $y = 100\%$ line, the better the performance of the corresponding algorithm.

Topologies. We use two kinds of topologies: the *Sparse* topologies are real topologies given to us by an ISP; the *Brite* topologies are synthetic topologies.

Each Sparse topology was obtained as follows: The operator of the ISP performed traceroute from a few end-hosts located inside her network toward a large number of external end-hosts; she discarded all incomplete paths. In this way, she collected a router-level graph (where each vertex corresponds to an IP router and each edge corresponds to an IP-level link). Moreover, she mapped each IP router to an Autonomous System (AS) and created an AS-level graph, where each vertex corresponds to a border router and each edge corresponds to either an inter-domain link between border routers of peering ASes, or an intra-domain path between two border routers of the same AS (see Section 4.1, scenario "The ISP curious about its peer"). In general, an ISP wants to monitor its peers at the AS level (it is not interested in each peer's internals), hence, we use the AS-level graph as the network topology. The router-level graph tells us how the links in the AS-level graph are correlated—if a router-level link becomes congested, then all the AS-level links that share this router-level link become congested at the same time.

Each Brite topology also consists of a router-level and an AS-level graph, each derived using the corresponding module of the Brite topology generator [Bri].

For the Correlation algorithm which relies on the Correlation Sets assumption, we define, for both kinds of topologies, one correlation set per AS, i.e., all links that belong to one AS are assigned to the same correlation set. In short, since we do not know which links of each AS are correlated, we assume that *all* links that belong to the same AS may be correlated. To define correlation sets in this manner, we need to map each link in the network graph to an AS, but no additional information, e.g., correlation factors between different links.

We show results for a representative Sparse topology of about 2000 links and a representative Brite topology of about 1000 links, each of them with 1500 paths—the results for other topologies were similar. Furthermore, the Identifiability++ condition holds only for the Brite topology.

Simulator. In the beginning of each experiment, we determine the probability that each (AS-level) link is congested and the degree of correlation be-

tween congested links (depending on whether they share underlying router-level links). In the experiments that we present here, a certain percentage of the links are assigned a positive congestion probability chosen at random between 0 and 1. Which particular links have a positive probability of congestion differs, depending on the scenario we are simulating.

Each experiment consists of multiple snapshots. In the beginning of each snapshot, we flip a biased coin for each link, to determine whether the link will be good or congested, such that we respect the individual and joint probabilities of congestion determined in the beginning of the experiment; if we determine that a link will be good (resp. congested) in this interval, we randomly assign to it a packet-loss rate between 0 and 0.01 (resp. 0.01 and 1), according to the loss model in [PQW03] (and similar to the loss models in [SQZ06, NT07a]). In each interval, packets are sent along each path; for each packet that arrives at a given link, we flip a biased coin to determine whether it will be dropped or not, such that we respect the packet-loss rate assigned to the link in the beginning of the interval.

Scenarios. We consider the following scenarios:

• *Random Congestion*: In this scenario, the links that have a positive congestion probability are chosen at random.

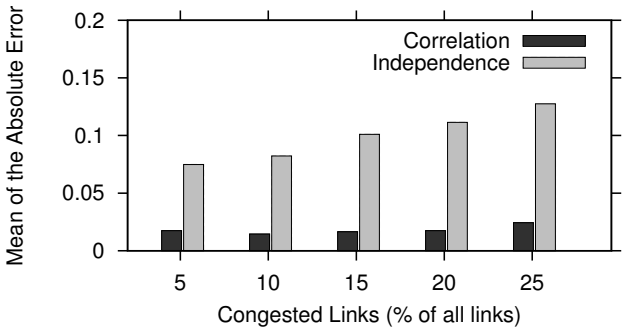
• *Concentrated Congestion*: In this scenario, the links that have a positive probability of congestion are chosen to be located toward the edge of the network, i.e., there is no congestion at the core.

• *No Independence*: In this scenario, the links that have a positive probability of congestion are chosen such that each of them is correlated with at least one other.

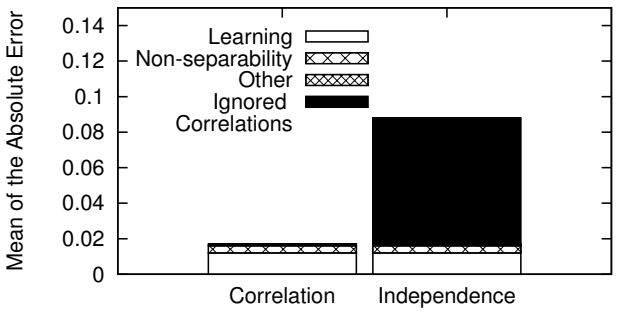
Performance when the Identifiability++ condition holds. We first look at the performance of the two algorithms when the Identifiability++ condition holds, i.e., the congestion probabilities of all correlation subsets are identifiable from end-to-end measurements.

In order to assess the benefits of taking into account link correlation, we simulate first the No Independence scenario for the Brite topology (Figure 5.3). For a fair comparison, we configure the Correlation algorithm to compute only the congestion probability of individual links, similar to the Independence algorithm.

First, we observe that the performance of the Correlation algorithm scales well as the percentage of links with a positive congestion probability increases



(a) Mean of the absolute error as the congestion level in the network increases.



(b) Breakdown of the mean of the absolute error when 10% of the links have a positive congestion probability.

Figure 5.3: Performance of the two algorithms in the No Independence scenario when the Identifiability++ condition holds. Both algorithms infer the congestion probability of individual links. Brite topology.

from 5% to 25%: for the Correlation algorithm, the mean of the absolute error stays below 0.03, while, for the Independence algorithm, the mean increases up to 0.14 (Figure 5.3(a)). Furthermore, the gap between the performance of the two algorithms widens as congestion increases, because more congestion implies that more correlated links are congested, and the Independence algorithm introduces a larger error by ignoring correlated links.

Second, we observe that, for the Independence algorithm, the error introduced by ignoring link correlation dominates (accounts for more than 80% of) the other factors. Figure 5.3(b) shows a breakdown of the mean of the absolute error for the two algorithms when 10% of the links have a positive congestion probability. One error factor is the insufficient "learning" of the congestion probabilities of the paths; this accounts for 70% of the mean absolute error for the Correlation algorithm and 14% for the Independence algorithm. We should note that this error factor can be mitigated by considering more snapshots per

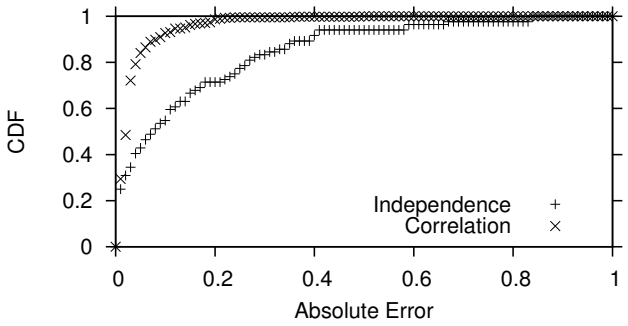


Figure 5.4: CDF of the absolute error in the No Independence scenario when the Identifiability++ condition holds. The Correlation algorithm infers the congestion probabilities of all correlation subsets that are potentially congested, whereas the Independence algorithm infers the congestion probability of all potentially congested links. Brite topology. 10% of the links have a positive congestion probability.

experiment (i.e., computing the congestion probabilities of links over longer periods of time), but that would also worsen the granularity of the results (we want to be able to compute the congestion probabilities of links over minutes or, perhaps, hours, but not days).

A smaller error factor is "non-separability": to determine whether a path is congested, we compare the packet-loss rate on that path to a threshold; hence, it is possible to mis-classify a congested path as good (or vice versa), which causes the Separability assumption to be violated and introduces noise in the measurements. We found all other error factors (e.g., the numerical error when solving the system of equations) to be negligible (below 1% of the mean of the absolute error).

Next, we look at the performance of the Correlation algorithm when computing the congestion probabilities of all correlation subsets (Figure 5.4). We consider again the *No Independence* scenario for the Brite topology when 10% of the links have a positive congestion probability. For the Correlation algorithm, we plot the CDF of the absolute error of the congestion probabilities of all correlation subsets that are potentially congested, whereas for the Independence algorithm, we plot the CDF of the absolute error of the congestion probabilities of all potentially congested links. In this case, the number of potentially congested correlation subsets is roughly three times larger than the number of potentially congested links. We observe that the Correlation algorithm significantly outperforms the Independence algorithm: The Correlation algorithm computes the congestion probability of 90% of the correlation subsets that are

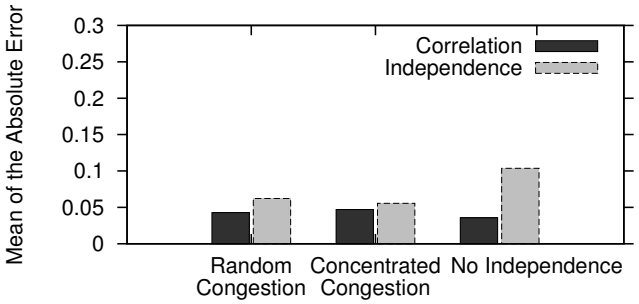


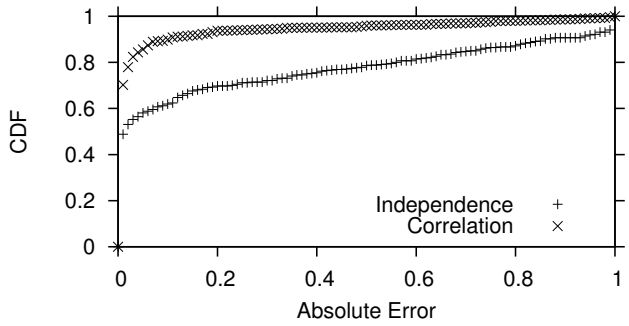
Figure 5.5: Mean of the absolute error in various scenarios when the Identifiability++ condition holds and the congestion probabilities of links change in time. Both algorithms infer the congestion probability of individual links. Brite topology. 10% of the links have a positive congestion probability.

potentially congested with an absolute error of less than 0.1, whereas the Independence algorithm infers the congestion probability of 90% of the potentially congested links with an absolute error of less than 0.4.

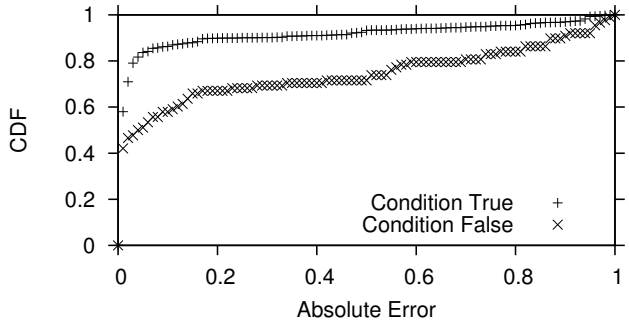
Finally, we look at the performance of the two algorithms when the congestion probabilities of links change in time. Figure 5.5 shows the mean of the absolute error in the Random Congestion, Correlated Congestion and No Independence scenarios. In this case, we configure the Correlation algorithm to compute only the congestion probability of individual links, similar to the Independence algorithm. We see that even under non-stationary network dynamics, the Correlation algorithm performs well with an absolute error below 0.7.

Performance when the Identifiability++ condition does not hold. We now look at the performance of the two algorithms when the Identifiability++ condition does not hold, i.e., we cannot identify from end-to-end measurements the congestion probabilities of all correlation subsets.

We consider the performance of the two algorithms for the Sparse topology, in the No Independence scenario. This topology contains large correlation sets, therefore, we configure the Correlation algorithm to compute the congestion probabilities of a maximum number of correlation subsets depending on our computational resource; these probabilities include also the congestion probabilities of all individual links. The Correlation algorithm computes roughly twice the number of probabilities inferred by the Independence algorithm. Figure 5.6(a) shows the CDF of the absolute error, when 10% of the links have a positive congestion probability. In this case as well, we see that the Correlation algorithm significantly outperforms the Independence algorithm: The Correla-



(a) CDF of the absolute error.



(b) CDF of the absolute error of the potentially congested correlation subsets for which the Identifiability++ condition holds and of the ones for which this condition does not hold.

Figure 5.6: Performance of the two algorithms in the No Independence scenario when the Identifiability++ condition does not hold. The Correlation algorithm infers the congestion probabilities of a maximum number of potentially congested correlation subsets depending on our computational resources, whereas the Independence algorithm infers the congestion probability of all potentially congested links. Sparse topology. 10% of the links have a positive congestion probability.

tion algorithm computes the congestion probability of 80% of the correlation subsets that are potentially congested with an absolute error of less than 0.1, whereas the Independence algorithm infers the congestion probability of 80% of the potentially congested links with an absolute error of less than 0.6. Furthermore, if we plot separately the CDF of the absolute error of the potentially congested correlation subsets for which the Identifiability++ condition holds and of the ones for which this condition does not hold, we see that the Correlation algorithm can compute accurately the congestion probabilities of the former, i.e., of the correlation subsets for which the Identifiability++ condition holds.

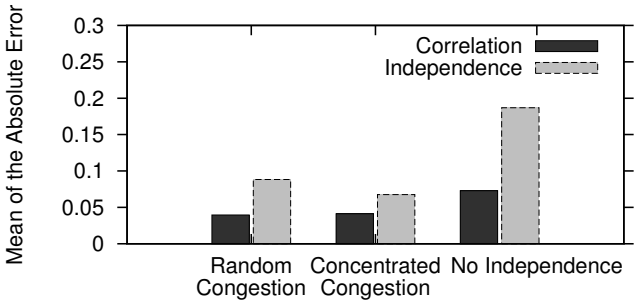


Figure 5.7: Mean of the absolute error in various scenarios when the Identifiability++ condition does not hold and the congestion probabilities of links change in time. Both algorithms infer the congestion probability of individual links. Sparse topology. 10% of the links have a positive congestion probability.

Finally, we also consider the case when the congestion probabilities of links change in time for the Sparse topology. Figure 5.5 shows the mean of the absolute error for the Sparse topology in the Random Congestion, Correlated Congestion and No Independence scenarios, when 10% of the links have a positive congestion probability. We configure the Correlation algorithm to compute only the congestion probability of individual links, similar to the Independence algorithm. We see that the Correlation algorithm performs as well as in the case of the Brite topology, whereas the Independence algorithm is much more inaccurate. The reason is that the Sparse topology involves large correlation sets, hence, two congested links are more likely to belong to the same correlation set. Since the Independence algorithm ignores these correlations, it learns the congestion probability badly.

5.3 A Practical Scenario

Our choice of a different loss tomography is motivated by practical evidence. We have talked to the operator of a European Tier-1 ISP (the "source ISP") who wanted to monitor the behavior and performance of its most important peers. In particular, for each peer, the operator of the source ISP wanted to understand: when the peer is responsible for performance problems encountered by the customers of the source ISP; how frequently the peer is congested and how its congestion level changes over the course of day or week; how well the peer reacts to exceptional situations like Border Gateway Protocol (BGP) failures, flash crowds, or distributed denial-of-service attacks. The source ISP does not have access to its peers' networks and cannot directly monitor their links; it can

only perform end-to-end measurements, i.e., monitor a number of paths from its own network to various Internet end-hosts. In this context, the operator of the source ISP asked us: can we apply network tomography to these end-to-end measurements to answer some or all of the above questions?

At first, this scenario sounded like a good match for Boolean loss tomography algorithms [PQW03, Duf06, DTDD07, NT07a], which monitor a set of paths during a snapshot and infer which particular links on these paths were congested during that snapshot. In order to assess weather Boolean loss tomography can provide trustworthy information in this scenario, we have obtained a real network topology from the operator of the source ISP on which we have tested these algorithms under various congestion patterns. The operator of the source ISP obtained this topology by performing traceroute from a few end-hosts located inside her network toward a large number of external end-hosts.

Unfortunately, Boolean loss tomography turned out to be too hard a problem in this scenario. State-of-the-art tomographic algorithms performed significantly worse than expected (Section 5.4), even when adjusted and fine-tuned to the scenario. Our initial reaction was to focus on the limitations of existing algorithms and design a new one that would overcome them; we found that each feature or twist we added to our algorithm to improve it came at the cost of significant complexity, yet brought little benefit—in the end, all algorithms that we tried performed very well under certain conditions (randomly congested links, link independence, stationary network dynamics, dense topologies) and equally badly under the opposite conditions which are more realistic. To conclude, in the scenario considered, we could not solve Boolean loss tomography with sufficient accuracy to be useful.

We argue that, in this scenario, the "right" problem to solve is Congestion Probability Inference, i.e., infer, for each set of links in the network, the probability that the links in this set are congested. This is less information than what would be provided by Boolean loss tomography: the source ISP learns only *how frequently* each set of links of each peer are congested, as opposed to *which particular* set of links of each peer are congested *when*. On the other hand, in practice, this information is more useful, because it can be obtained accurately under weaker assumptions and more challenging network conditions.

5.4 Limitations of Boolean Loss Tomography

In this section, we show that even though Congestion Probability Inference shares the same foundation with Boolean loss tomography, it does not inherit the limitations of the latter.

5.4.1 Boolean Loss tomography is ill-posed

The goal of Boolean loss tomography is to determine the set of congested links during a snapshot from the set of congested paths during that snapshot [Duf06]. This problem is ill-posed: for all network graphs[3], given the set of congested paths, there may be multiple possible solutions (sets of congested links) that could have led to this outcome. For example, in Figure 5.1, suppose that all three paths are congested during a snapshot; there are 8 possible sets of links that if congested would have led to this outcome: $\{e_1, e_4\}$, $\{e_1, e_2\}$, $\{e_3, e_4\}$, $\{e_1, e_3, e_4\}$, $\{e_1, e_2, e_3\}$, $\{e_1, e_2, e_4\}$, $\{e_2, e_3, e_4\}$, and $\{e_1, e_2, e_3, e_4\}$.

Since Boolean loss tomography is an ill-posed problem, no algorithm can solve it exactly, that is, identify the congested links without false negatives or positives for all possible sets of congested paths (Lemma 5.6). Nevertheless, it is possible to compute an approximation of the set of congested links that is close to the actual solution when certain additional assumptions hold. Therefore, what distinguishes different Boolean loss tomography algorithms from one another is the set of additional assumptions that each of them relies on.

Lemma 5.6. *In any network where the topology is not the complete graph, Boolean loss tomography is ill-posed.*

Proof. We will show that there is no network where the set of congested links is identifiable for all possible sets of congested paths, with the exception of the trivial network in which the topology is a complete graph, i.e., hosts are directly interconnected.

If the topology is not the complete graph, there is at least one node which acts as a router. We denote by $\mathcal{E}_{in}$ the set of ingress links of this router, and by $\mathcal{E}_{out}$ the set of its egress links. In the network tomography model (Section 2.1), links are unidirectional , thus, $\mathcal{E}_{in} \cap \mathcal{E}_{out} = \emptyset$. Furthermore, since a path must start and terminate at a host, all paths that enter the router must also exit the router, hence, $Paths(\mathcal{E}_{in}) = Paths(\mathcal{E}_{out})$.

[3]Except for the trivial case when end-hosts are directly interconnected.

Suppose that the set of congested paths is $Paths(\mathcal{E}_{in})$. In this case, there are at least two possible sets of links, namely, $\mathcal{E}_{in}$ and $\mathcal{E}_{out}$ that if congested would have led to this outcome. Therefore, there is no one-to-one mapping between the set of congested links and the set of congested paths available from end-to-end measurements. In conclusion, we cannot identify the set of congested links for all possible sets of congested paths. □

5.4.2 Analysis of Tomographic Algorithms

In this section, we analyze three state-of-the-art algorithms which attempt to solve the Boolean loss tomography problem for mesh networks: (i) *Sparsity* (originally called Tomo[4]) [DTDD07], an adaptation of Duffield's inference algorithm for trees [Duf06] to mesh networks; (ii) *Bayesian-Independence* (originally called CLINK) [NT07a]; and (iii) *Bayesian-Correlation*, a new algorithm that we developed for this work [GAT11]. We experimentally show that neither of them performs accurate inference in the practical scenario described in Section 5.3. Our point is not that these algorithms are not good (we pick them precisely because they represent the state-of-the-art). Instead, we argue that any Boolean loss tomography algorithm is bound to be accurate in some scenarios and inaccurate in others, and there is no evidence that the scenarios favored by one algorithm occur more frequently than those favored by the others.

The Bayesian algorithms (Bayesian-Independence and Bayesian-Correlation) attempt to solve the Boolean loss tomography problem by using the congestion probabilities of links or respectively, of correlation subsets as prior information. Toward this goal, they pose Boolean loss tomography as a Maximum Likelihood Estimation (MLE) problem: of all the possible solutions, where a solution consists of the set of congested links, it looks for the one that occurred with the highest probability.

Intuition

First, we explain through toy examples the sources of inaccuracy introduced by each algorithm.

Sparsity[DTDD07]. The gist behind this algorithm is that a few congested links are responsible for many congested paths. Under the assumption that all links are equally likely to be congested, i.e., the Link Homogeneity assumption

[4]We use new names for the existing algorithms, in order to better distinguish them from each other.

introduced in Section 2.5, Sparsity "favors" links that participate in a higer number of congested paths, i.e., the larger the number of congested paths in which a link participates, the more likely it is to be labeled as congested. For example, in the toy topology of Figure 5.1, if all three paths are congested, Sparsity will infer that the congested links are $\{e_1, e_4\}$ because each of them participates in two congested paths whereas each of the other links participate in only one congested path.

Sparsity works best in scenarios where congestion is concentrated in a few links. This is not the case, for instance, when there exists a lot of congestion at the edge of the network, i.e., many links adjacent to end-hosts are congested at the same time. For example, in Figure 5.1, if links e_3 and e_4 are both congested, which will cause all paths to be congested, and Sparsity will pick solution $\{e_1, e_4\}$, i.e., it will miss one congested link and falsely blame one good link.

Bayesian-Independence[NT07a]. This algorithm consists of two steps: (i) *Congestion Probability Inference*, which monitors the network and learns the probability with which each solution occurs, and (ii) *Bayesian Inference*, which looks at the status of paths during each snapshot and determines which set of links were most likely congested during that snapshot, based on the output of the previous step. For example, in Figure 5.1, if all paths are congested, Bayesian-Independence will consider all 8 possible solutions and pick the one that occurs with the highest probability.

The Congestion Probability Inference step monitors the congestion statuses of paths, learns the probability that each set of paths is congested and, from these, under the Link Independence assumption, computes the probability that each link is congested. We illustrate with the example of Figure 5.1: First, the method computes the probability that path p_1 is good, which is equal to the probability that link e_1 and link e_3 are both good, and forms the first equation in Equation 4.1. In the same way, it computes the probability that each path and each pair of paths is good and forms the remaining equations in Equation 4.1. The resulting system has four unknowns (one for each link) and four linearly independent equations, hence, gives us the probability that each link is good. Assuming Link Independence, we can easily compute the probability of each particular solution, e.g., the probability of solution $\{e_1, e_3\}$ is $\mathbb{P}(Z_{e_1} = 1)\ \mathbb{P}(Z_{e_2} = 0)\ \mathbb{P}(Z_{e_3} = 1)\ \mathbb{P}(Z_{e_4} = 0)$.

Bayesian-Independence needs the Link Independence assumption, in order to form equations by combining probabilities related to different links. As

previously pointed out in Section 4.1, this assumption does not always hold in practice, which causes Bayesian-Independence to compute some probabilities incorrectly, leading to incorrect inference. For example, suppose that, in Figure 5.1, links e_1 and e_2 are always good, while e_3 and e_4 are perfectly correlated (either both are congested or both are good). This means that $\mathbb{P}(Z_{e_3} = 0, Z_{e_4} = 0) \neq \mathbb{P}(Z_{e_3} = 0)\ \mathbb{P}(Z_{e_4} = 0)$, and the last two equations in Equation 4.1 are wrong. As a result, Bayesian-Independence incorrectly determines that $\{e_1, e_4\}$ is the solution with the highest probability and always picks it over the correct one $\{e_3,\ e_4\}$.

A more subtle source of inaccuracy in the Bayesian Inference step is the following: Bayesian-Independence determines whether link e_j was congested during a *particular* snapshot based on the probability that link e_j is congested during *any* snapshot. More formally, under the Stationarity assumption, Bayesian-Independence uses as estimate for the value of random variable Z_{e_j} its expected value $\mathbb{E}[Z_{e_j}] = \mathbb{P}\left(Z_{e_j} = 1\right)$. We illustrate with an example. Suppose that the Congestion Probability Inference module observes the network in Figure 5.1 for an hour and determines the following probabilities:

$$\mathbb{P}(Z_{e_1} = 1, Z_{e_2} = 0, Z_{e_3} = 0, Z_{e_4} = 0) = 0.3.$$
$$\mathbb{P}(Z_{e_1} = 1, Z_{e_2} = 0, Z_{e_3} = 1, Z_{e_4} = 0) = 0.1.$$

This means that, during the one hour of monitoring, $\{e_1\}$ was the only congested link in the network for 30% of the time, while $\{e_1, e_3\}$ were the only congested links in the network for 10% of the time. Now suppose that during the last 1-minute interval within this hour, the congested paths are $\{p_1, p_2\}$; the Bayesian Inference module determines that there are two possible solutions for this interval, $\{e_1\}$ and $\{e_1, e_3\}$, and picks the first one (because it has a higher probability associated with it). In essence, Bayesian Inference determines that this solution is more likely to have occurred *during the last minute*, because it occurred more frequently *over the last hour*.

In practice, we cannot tell whether this estimation is valid, unless we have "insider information" on network conditions. For example, consider a link that is normally congested very rarely, and the Congestion Probability Inference step correctly computes a low congestion probability for it; suppose this link incurs a technical failure or comes under a flooding attack and becomes severely congested for a few time intervals; unless we already know when this failure/attack occurs and how long it lasts, Bayesian Inference will not pick this link as congested as it has a low congestion probability associated with it. So, even if

Congestion Probability Inference correctly computes for what fraction of time a link is congested, Bayesian Inference cannot use this information correctly because it inevitably picks the expected value of a random variable for its value.

Finally, the Bayesian Inference step is an NP-complete problem, hence, Bayesian-Independence uses an approximate algorithm to pick the solution that occurred with the highest probability, which means that it may not always pick the right one.

To summarize, Bayesian-Independence introduces three additional sources of inaccuracy: the Link Independence assumption (used in both steps), the fact that it approximates the value of the random variable Z_{e_j} with its expected value (in the Bayesian Inference step), and the use of an approximate algorithm to pick the solution that occurred with the highest probability (also in the Bayesian Inference step).

Bayesian-Correlation [GAT11]. In an effort to remove one source of inaccuracy, we developed a new algorithm that takes into account link correlations. It is similar to Bayesian-Independence , i.e., it also consists of a Congestion Probability Inference and a Bayesian Inference step, however, instead of the Link Independence assumption, it relies on the Correlation Sets assumption. In the Congestion Probability Inference step, we use our algorithm described in Section 5.2.4. For instance, in the example of Figure 5.1, it treats $\mathbb{P}(Z_{e_3} = 0, Z_{e_4} = 0)$ as an extra unknown, as opposed to mistakenly breaking it into $\mathbb{P}(Z_{e_3} = 0)\ \mathbb{P}(Z_{e_3} = 0)$, and forms the equations in Figure 4.2. The resulting system has 5 unknowns (one for each link plus one for the pair of correlated links $\{e_3, e_4\}$) and 5 linearly independent equations. Hence, we can determine these 5 probabilities by solving the system of equations, and compute the probability of each solution, e.g., the probability of solution $\{e_1, e_3\}$ is $\mathbb{P}(Z_{e_1} = 1)\ \mathbb{P}(Z_{e_2} = 0)\ \mathbb{P}(Z_{e_3} = 1, Z_{e_4} = 0)$.

Nevertheless, taking link correlations into account comes at the price of introducing extra unknowns, and we can compute all of them if and only if the Identifiability++ condition holds. For instance, in the example of Figure 4.6, it is impossible to compute the probability that $\{e_1\}$ is good or the probability that $\{e_2, e_3\}$ are both good. The intuition is the following: both these sets of links are traversed by the same set of paths $\{p_1, p_2\}$; this makes it impossible to distinguish one pair from the other based on path observations and to compute the probability that each pair is good. So, the Congestion Probability Inference step of Bayesian-Correlation cannot always compute the probability of all solutions, because the the Identifiability++ condition does not always hold. As a

	Sparsity	Bayesian-Independence		Bayesian-Correlation	
		Step 1	Step 2	Step 1	Step 2
Routing Stability	×	×	×	×	×
Link Identifiability	×	×	×		
Stationarity		×	×	×	×
Separability	×	×	×	×	×
Link Homogeneity	×				
Link Independence	×	×	×		
Correlation Sets				×	×
Identifiability++				×	×
Other approx./ heuristic	×		×		×

Table 5.4: Sources of inaccuracy for Boolean loss tomography algorithms: assumptions, conditions, and approximations/heuristics.

result, the Bayesian Inference step does not have all the information it needs to pick the likeliest solution.

To summarize, Bayesian-Correlation introduces three additional sources of inaccuracy: the Correlation Sets assumption and the Identifiability++ condition (used in both steps) and—like Bayesian-Independence—the fact that it approximates the values of random variables with their expected values and the use of an approximate algorithm to pick the solution that occurred with the highest probability (in the Bayesian Inference step).

Conclusion. We have seen that each algorithm introduces its own sources of inaccuracy (summary in Table 5.4), and there is no basis for arguing that one algorithm covers more cases than the others.

Experiments

We now look at the performance of the three algorithms under various scenarios (Figure 5.8). We assume that Routing Stability, Separability, and Correlation Sets always hold, because this is the weakest set of assumptions under which we can solve Boolean loss tomography; the rest of the assumptions and conditions in Table 5.4 may or may not hold, depending on the scenario.

We use the same simulator, topologies and scenarios as the ones described in Section 5.2.5. In all scenarios, 10% of the links have a positive congestion probability.

Metrics. We consider two metrics: during a particular time interval, the *detection rate* of an algorithm is the fraction of congested links that the algorithm correctly identified as congested; the *false positive rate* of an algorithm is the fraction of links incorrectly identified as congested out of all links inferred

as congested by the algorithm. Each detection rate and false-positive rate we show is an average over 1000 time intervals.

Random Congestion (Brite). As we see in Figure 5.8, all Inference algorithms perform equally well: on average, they identify 90% of the congested links and miss fewer than 2% of them (except for Bayesian-Independence which misses 10%).

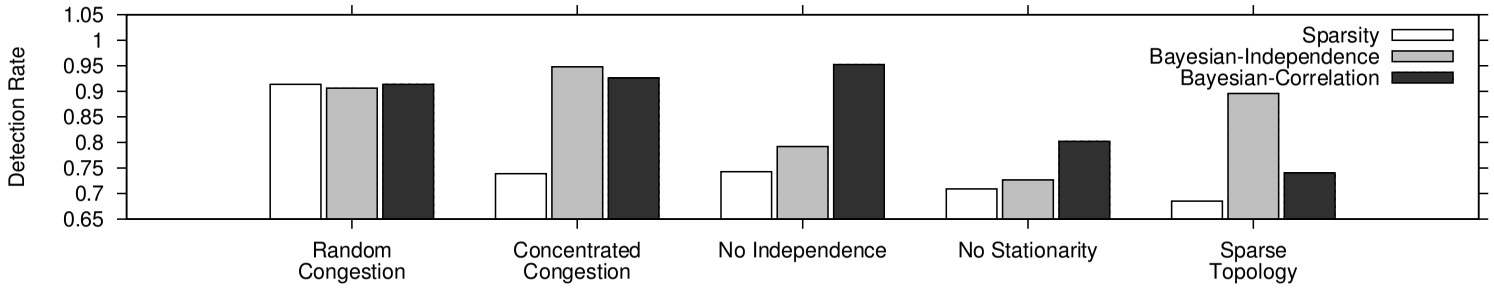
The intuition is the following. The Brite topology models a full AS-level topology, hence, it is relatively "dense," i.e., paths tend to criss-cross. This is good for Inference algorithms, because the denser the topology, the fewer the possible solutions to each observation—which means that the heuristic/approximate aspect of each algorithm is exercised less. Bayesian-Independence performs slightly worse, because it assumes that links are independent, whereas, during several snapshots, some of the congested links happen to be correlated (share an underlying router-level link).

Concentrated Congestion (Brite). As we see in Figure 5.8, Sparsity's detection rate drops to 75%, while its false-positive rate rises to 10%. This happens because Sparsity assumes Link Homogeneity and picks links that are traversed by many congested paths, hence it is more likely to pick solutions that involve links located close to the core of the network. This result does not imply that Sparsity is worse than the other algorithms—just that it performs worse in this particular scenario.

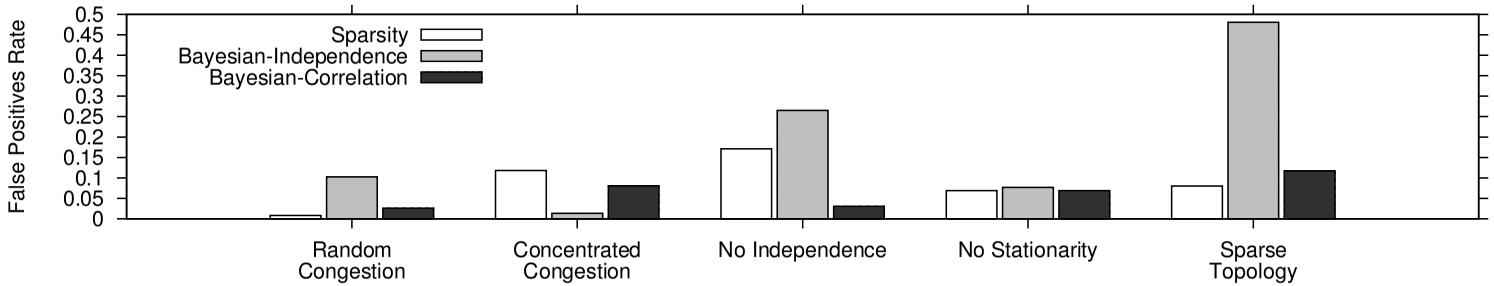
No Independence (Brite). As we see in Figure 5.8, Bayesian-Independence's detection rate drops below 80%, while its false-positive rate rises to 25%; this happens because its Congestion Probability Inference step assumes Link Independence, hence, learns the probability of each set of links incorrectly.

No Stationarity (Brite). This scenario is similar to the No Independence scenario, plus the congestion probabilities of links change in time. As we see in Fig. 5.8, it is the turn of Bayesian-Correlation's detection rate to drop below 80%; this happens because its Bayesian Inference step assumes that a solution is more likely to have occurred during the last time interval, just because it occurred more frequently throughout the entire experiment.

Sparse Topology. This scenario is the Random Congestion scenario applied to the Sparse topology. As we see in Figure 5.8, all Inference algorithms suffer. The fact that Bayesian-Independence has a 90% detection rate should not be mistaken for success: it achieves this by aggressively marking links as congested, which results in a 45% false-positive rate.



(a) Detection Rate.



(b) False Positive Rate.

Figure 5.8: Performance of Boolean loss tomography algorithms under various realistic congestion scenarios, when 10% of the network links have a positive probability of being congested. The Random Congestion, Concentrated Congestion, No Independence and No Stationarity scenarios are simulated on the Brite topology. The Identifiability++ condition holds only for the Brite topology.

The intuition is the following. The Sparse topology was created by running traceroute from the source ISP to various Internet end-hosts. However, most traceroutes returned incomplete/inconclusive results and had to be discarded, which resulted in a "sparse" view, where few paths intersect one another. This is bad for Boolean loss tomography algorithms, because the sparser the topology, the less information we obtain about the link characteristics from end-to-end measurements—which means that each algorithm has to rely more on its heuristic/approximate aspect to pick a solution. Note that we did not in any way engineer this scenario to make the algorithms fail as we did in the previous scenarios—we did not introduce extra link correlations or non-stationarity.

We should clarify that the Sparse topology is the most complete topology that the ISP operator was able to collect with the resources (the monitoring points) she had at her disposal. One might argue that, if the operator had done a better job and collected a more complete (less sparse) topology, the algorithms would have performed better. This is true, however, in our experience from working with the operator, piecing together a topology from traceroutes is a complex task—some routers respond to a traceroute probe through a different interface than the one where the probe was received, some routers do not respond to traceroute probes at all, while load-balancing interferes with traceroute results. Hence, we think it is fair to assume that operators are typically not able to collect complete topologies.

Conclusion. Because of the ill-posed nature of Boolean loss tomography, any Boolean loss tomography algorithm can perform badly under certain network conditions, and there is no evidence that such conditions do not occur in practice. Moreover, all algorithms perform badly on Sparse topologies—in particular, each algorithm performs worse on Sparse topologies under easy conditions (random congestion) than on Brite topologies under worst-case conditions (congestion at the edges for Sparsity, link correlations for Bayesian-Independence, and non-stationarity for Bayesian-Correlation).

5.5 Why a Different Loss Tomography?

In this section, we argue that it makes more sense to solve Congestion Probability Inference rather than Boolean loss tomography.

Ideally, Boolean loss tomography provides the source ISP with the congestion status of each link in the network during each snapshot. This information would enable the source ISP to attribute blame to a peer for a particular connec-

tivity/performance problem faced by the source ISP's customers and/or request compensation in case a Service Level Agreement (SLA) has been violated. However, we have seen in Section 5.4 that for the Sparse topology, state of the art Boolean loss tomography algorithms yield a detection rate as low as 68% and a false positive rate as high as 47%; attributing blame or extracting compensation is practically impossible based on this level of accuracy.

Congestion Probability Inference provides less information than Boolean loss tomography: if accurately solved, it would provide the source ISP with the congestion probability of each set of links in the network, i.e., *how frequently* each set of links are congested, but not *which particular* links were congested *when*.

On the other hand, our algorithm described in Section 5.2.4 (Step 1 of Bayesian-Correlation) solves Congestion Probability Inference with fewer sources of inaccuracy than Boolean loss tomography algorithms:

- It assumes Routing Stability, Stationarity, Separability, and Correlation Sets—a weaker set of assumptions than those assumed by Sparsity and Bayesian-Independence.

- Unlike the Bayesian algorithms (Bayesian-Independence and Bayesian-Correlation), our algorithm does not need to solve a NP complete problem.

- Unlike the Bayesian algorithms, our algorithm does not need to approximate the value of random variables with its expected values: if we compute that $\mathbb{P}\left(Z_{e_j} = 1\right) = 0.3$ over N snapshots, we interpret this as "e_j was congested for 30% of the N snapshots." In contrast, the Bayesian algorithms use the same information to infer during which particular snapshot e_j was congested. When network conditions change over time, the Bayesian algorithms may make the wrong decision as discussed in Section 5.4; our result, however, still holds, because it concerns the *average* behavior of the link over the N snapshots, and not the diagnosis of the congested links over a single snapshot.

5.6 Conclusion

In this chapter, we have argued for a different loss tomography, namely, Congestion Probability Inference that computes the congestion probability of each set of links, that is, it determines how often all links belonging to each set are congested. Unlike continuous or Boolean loss tomography that are ill-posed, Congestion Probability Inference is well-posed under certain well-defined con-

ditions. Consequently, tomographic algorithms solving this problem can work under weaker assumptions than those required by algorithms solving one of the traditional versions of loss tomography (either continuous or Boolean loss tomography). We have designed an algorithm solving Congestion Probability Inference under the weakest set of assumptions made by any tomographic algorithm to date. We have shown that in the scenario of an ISP that wants to monitors the performance of its peers, our algorithm provides trustworthy information under challenging conditions such as sparse topologies and non-stationary network dynamics.

Symbol	Definition
$\widehat{E}$	the set of all potentially congested links
$\mathcal{S}_k \in S, \mathcal{S}_k \subseteq \widehat{E}$	a potentially congested correlation subset
$\widehat{S}$	an ordering of all potentially congested correlation subsets
$\overline{\mathcal{S}}_k$	the complement of $\mathcal{S}_k \in \widehat{S}$
$\mathcal{P} \subseteq P$	a path set
$\widehat{\mathcal{P}}$	an ordering of path sets
$Paths(\mathcal{S}_k)$	all paths traversing links in $\mathcal{S}_k$
$Links(\mathcal{P})$	all links traversed by paths in $\mathcal{P}$
$\widehat{Links}(\mathcal{P})$	all potentially congested links traversed by paths in $\mathcal{P}$
$\boldsymbol{\alpha}_{\mathcal{P},\mathcal{S}_k} = 1$	$\widehat{Links}(\mathcal{P}) \cap \mathcal{C}_p = \mathcal{S}_k$, where $\mathcal{C}_p$ is the correlation set of $\mathcal{S}_k$
$Row(\mathcal{P}, \widehat{S})$	a row vector where each element corresponds to $\boldsymbol{\alpha}_{\mathcal{P},\mathcal{S}_k}$, with $\mathcal{S}_k \in \widehat{S}$
$Matrix(\widehat{\mathcal{P}}, \widehat{S})$	a matrix where each row corresponds to $Row(\mathcal{P}, \widehat{S})$, with $\mathcal{P} \in \widehat{\mathcal{P}}$
$\mathcal{P}_{\overline{\mathcal{C}}_p}$	the paths in $\mathcal{P}$ which do not traverse any potentially congested link in correlation set $\mathcal{C}_p$
$\mathcal{P}_{\overline{\mathcal{S}}_k}$	the paths in $\mathcal{P}$ which traverse links in $\overline{\mathcal{S}}_k$, the complement of correlation subset $\mathcal{S}_k \in \widehat{S}$.
$\mathcal{P}_{\mathcal{S}_k}$	the paths in $\mathcal{P}$ which traverse links in correlation subset $\mathcal{S}_k \in \widehat{S}$, but do not traverse links in $\overline{\mathcal{S}}_k$.
$\Omega_{\mathcal{P},\mathcal{S}_k}$	the set of all path sets $\mathcal{Q} \subseteq \mathcal{P}_{\mathcal{S}_k}$ such that $\boldsymbol{\alpha}_{\mathcal{Q},\mathcal{S}_k} = 1$

Table 5.5: Symbols used in Chapter 5.

Chapter 6

Conclusion

In this thesis, we have strived to bring network loss tomography closer to practice. Toward this goal, we have designed tomographic algorithms that work under assumptions weaker than those required by state-of-the-art algorithms.

Our first contribution is in the context of continuous loss tomography, we have proposed Netscope, an algorithm that infers the loss rates of network links from end-to-end measurements (Chapter 3). Inspired by previous work [NT07b], we have designed an algorithm that gains initial information about the network by computing the variances of the loss rates of links, and by using these variances as an indication of the congestion level of links, i.e., the more congested the link, the higher the variance of its loss rate. Its novelty lies in the way it uses this information—to identify and characterize the maximum set of links whose loss rates can be accurately inferred from end-to-end measurements. We have shown that our algorithm performs significantly better than the alternatives, and that this advantage increases with the number of congested links in the network. Furthermore, Netscope is robust in the sense that it requires no parameter tunning. We validated Netscope's performance by using PlanetLab experiments: We have built a "Internet tomographer" that runs on PlanetLab nodes and infers the loss rates of links located between them; we have used some of the measured paths for inference and others for validation, and we have shown that the results are consistent.

Second, we have shown that it is feasible to perform network loss tomography in the presence of "link correlations," i.e., when the losses that occur on one link depend on the losses that occur on other links in the network (Chapter 4). More precisely, we have formally derived the necessary and sufficient condition under which the probability that each set of links is congested is statistically identi-

fiable from end-to-end measurements even in the presence of link correlations. In doing so, we have challenged one of the popular assumptions in network loss tomography, specifically, the assumption that all links are independent. The model we have proposed assumes we know which links are most likely to be correlated, but it does not assume any knowledge about the nature or the degree of their correlation. In practice, we consider that all links in the same local area network or the same administrative domain are potentially correlated, because they might be sharing physical links, network equipment, or even management processes.

Finally, we have designed a practical algorithm that solves "Congestion Probability Inference" even in the presence of link correlations, i.e., our algorithm infers the probability that each set of links is congested under the link correlation model proposed in Chapter 4 (Chapter 5). We modeled Congestion Probability Inference as a system of linear equations where each equation corresponds to a set of paths. Because it is infeasible to consider an equation for each set of paths in the network, our algorithm finds the maximum number of linearly independent equations by selecting particular sets of paths based on our theoretical results. On the one hand, the information provided by our algorithm is less than that provided by the existing alternatives that infer either the loss rates or the congestion statuses of links, i.e., we only learn how often each set of links is congested, as opposed to how many packets were lost at each link, or to which particular links were congested when. On the other hand, this information is more useful in practice because our algorithm works under assumptions weaker than those required by the existing alternatives, and we experimentally show that it is accurate under challenging network conditions such as non-stationary network dynamics and sparse topologies.

References

[ABF+00] Andrew Adams, Tian Bu, Timur Friedman, Joseph Horowitz, Don Towsley, Ramon Caceres, Nick Duffield, Francesco Lo Presti, Sue B. Moon, and Vern Paxson. The Use of End-to-end Multicast Measurements for Characterizing Internal Network Behavior. *IEEE Communications Magazine*, May 2000.

[AdVE07] Dogou Arifler, Gustavo de Veciana, and Brain L. Evans. A factor analysis approach to inferring congestion sharing based on flow level measurements. *IEEE/ACM Transactions on Networking*, 2007.

[BDPT02] T. Bu, N. Duffield, F. Lo Presti, and D. Towsley. Network Tomography on General Topologies. In *Proceedings of the ACM SIGMETRICS Conference*, 2002.

[BMT05] Alexandros Batsakis, Tanu Malik, and Andreas Terzis. Practical passive lossy link inference. In *Proc. of PAM 2005*, 2005.

[Bri] Boston University Representative Internet Topology Generator. `http://www.cs.bu.edu/brite/`.

[CCL+04] Rui Castro, Mark Coates, Gang Liang, Robert Nowak, and Bin Yu. Network tomography: recent developments. *Statistical Science*, 19:499–517, 2004.

[CDHT99] R. Caceres, N. G. Duffield, J. Horowitz, and D. Towsley. Multicast-based Inference of Network-Internal Loss Characteristics. *IEEE Transactions on Information Theory*, 45:2462–2480, 1999.

[CHRY02] M. Coates, A. Hero, R.Nowak, and B. Yu. Internet tomography. *IEEE Signal Processing Magazine*, 19, May 2002.

[CN00] M. Coates and R. Nowak. Network Loss Inference Using Unicast End-to-End Measurement. In *Proceedings of the ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, 2000.

[DHT+] Nick Duffield, Joseph Horowitz, Don Towsley, Wei Wei, and Timur Friedman. Multicast-based loss inference with missing data.

[DPPT01] N.G. Duffield, F. Lo Presti, V. Paxson, and D. Towsley. Inferring Link Loss Using Striped Unicast Probes. In *Proceedings of the IEEE INFOCOM Conference*, 2001.

[DTDD07] Amogh Dhamdhere, Reneta Teixeira, Constantine Drovolis, and Christophe Diot. Netdiagnoser: Troubleshooting network unreachabilities usind end-to-end probes and routing data. In *Proceedings of ACM Conext*, 2007.

[Duf06] N. G. Duffield. Network Tomography of Binary Network Performance Characteristics. *IEEE Transactions on Information Theory*, 52(12):5373–5388, December 2006.

[GAT11] D. Ghita, K. Argyraki, and P. Thiran. Inference algorithms analysis. Technical report, EPFL, 2011.

[GL96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.

[HBB00] K. Harfoush, A. Bestavros, and John Byers. Robust identification of shared losses using end-to-end unicast probes. In *Proc. of ICNP'00*, 2000.

[Jac89] V. Jacobson. traceroute, ftp://ftp.ee.lbl.gov/traceroute.tar.z, 1989.

[min] Multicast-based inference of network-internal characteristics. `http://www-net.cs.umass.edu/minc/`.

[MSWA03] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. User-level Internet Path Diagnosis. In *ACM SOSP*, 2003.

[NGK+09] Hung Nguyen, Denisa Ghita, Maciej Kurant, Katerina Argyraki, and Patrick Thiran. Fundamental Properties of Routing Matrices and Their Implications for Network Tomography. Technical report, EPFL, February 2009. Available (for Infocom 2010 review only) at `http://icwww.epfl.ch/~argyraki/RoutingMatrixRank.pdf`.

[NT07a] H. X. Nguyen and P. Thiran. The Boolean Solution to the Congested IP Link Location Problem: Theory and Practice. In *Proceedings of the IEEE INFOCOM Conference*, 2007.

[NT07b] Hung X. Nguyen and Patrick Thiran. Network Loss Inference with Second Order Statistics of End-to-End Flows. In *Proceedings of the IEEE Internet Measurement Conference (IMC)*, 2007.

[Pla] PlanetLab: An Open Platform for Developing, Deploying, and Accessing Planetary-scale Services. `http://www.planet-lab.org/`.

[PQW03] V. N. Padmanabhan, L. Qiu, and H. J. Wang. Server-based Inference of Internet Performance. In *Proceedings of the IEEE INFOCOM Conference*, 2003.

[RKT02] Dan Rubenstein, Jim Kurose, and Don Towsley. Detecting shared congestion of flows via end-to-end measurement. *IEEE/ACM Transactions on Networking*, 10(3), June 2002.

[SB07] Joel Sommers and Paul Barford. An active measurement system for shared environments. In *Proceedings of ACM SIGCOMM IMC*, October 2007.

[SQZ06] Han Hee Song, Lili Qiu, and Yin Zhang. NetQuest: A Flexible Framework for Large-Scale Network Measurement. In *Proceedings of the ACM SIGMETRICS Conference*, 2006.

[SWA03] N. Spring, D. Wetherall, and T. Anderson. Scriptroute: A Public Internet Measurement Facility. In *USITS*, 2003.

[TCN01] Yolanda Tsang, Mark Coates, and Robert Nowak. Passive Network Tomography Using the EM Algorithms. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2001.

[Var96] Y. Vardi. Network Tomography: Estimating Source-Destination Traffic Intensities. *Journal of the American Statistical Association*, 91:365–377, 1996.

[V.P96] V.Paxson. End-to-End Routing Behaviour in the Internet. In *ACM SIGCOMM*, 1996.

[ZC07] Yao Zhao and Yan Chen. A suite of schemes for user-level network diagnosis without infrastructure. In *Proceedings of IEEE INFOCOM*, Anchorage, 2007.

[ZCB06] Yao Zhao, Yan Chen, and David Bindel. Toward Unbiased End-to-End Network Diagnosis. In *Proceedings of the ACM SIGCOMM Conference*, 2006.

[ZDPS01] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker. On the Constancy of Internet Path Properties. In *ACM SIGCOMM Internet Measurement Workshop*, 2001.

Appendix A

Congestion Probability Inference

A.1 A Heuristic to Speed Up the Algorithm

We can speed up Algorithm 5.1 by using the following heuristic: instead of considering all possible subsets of path set $Paths(\mathcal{S}_k) \setminus Paths(\overline{\mathcal{S}}_k)$ as described in line 9 of Algorithm 5.1, we will consider only some of these subsets, namely, the ones more likely to generate a row which will increase the rank of the system's matrix. Given a subset $\mathcal{S}_k \in \widehat{S}$, we know that the row generated by path set $\mathcal{P} = Paths(\mathcal{S}_k) \setminus Paths(\overline{\mathcal{S}}_k)$ is already in the system's matrix because of the initial phase in lines 1-4 of Algorithm 5.1. The intuition is that in order to obtain a new linearly independent row, it is sufficient to slightly "disturb" this path combination by removing some of the paths in $\mathcal{P}$. Our heuristic (Algorithm A.1) considers all potentially congested correlation subsets $\mathcal{S}_l \in \widehat{S}$ which are covered by the paths in $\mathcal{P}$ (line 10), and checks if the path set $\mathcal{P} \setminus Paths(\mathcal{S}_l)$ satisfies the necessary condition (lines 11-13). If this test fails, it also checks whether removing from $\mathcal{P}$ the paths which traverse individual edges in $\mathcal{S}_l$ will generate a linearly independent row (lines 16-21).

In practice, our heuristic was always able to find the maximum number of linearly independent rows. However, we do not have a theoretical result which shows that this heuristic is complete.

Algorithm A.1 *Heuristic for Selection of Path Sets*

Input: $\widehat{S}$: a list of potentially congested correlation subsets
Variables: $\widehat{\mathcal{P}}$: a list of path sets
$\mathcal{P}$: a path set
$\mathcal{S}_k$: a correlation subset

1: $\widehat{\mathcal{P}} \leftarrow \langle\rangle$
2: **for all** $\mathcal{S}_k \in \widehat{S}$ **do**
3: $\quad \mathcal{P} \leftarrow \mathit{Paths}(\mathcal{S}_k) \setminus \mathit{Paths}(\overline{\mathcal{S}}_k)$
4: $\quad \widehat{\mathcal{P}} \leftarrow \widehat{\mathcal{P}} + \mathcal{P}$

5: $\mathbf{A} \leftarrow \mathit{Matrix}(\widehat{\mathcal{P}}, \widehat{S})$
6: $\mathbf{N} \leftarrow \mathit{NullSpace}\,(\mathbf{A})$

7: **repeat**
8: $\quad$ **for all** $\mathcal{S}_k \in \mathit{SortByHammingWeight}(\widehat{S}, \mathbf{N})$ **do**
9: $\quad\quad \mathcal{P} \leftarrow \mathit{Paths}(\mathcal{S}_k) \setminus \mathit{Paths}(\overline{\mathcal{S}}_k)$

10: $\quad\quad$ **for all** $\mathcal{S}_l \in \widehat{S}$ such that $\mathcal{S}_l \subseteq \widehat{\mathit{Links}}(\mathcal{P})$ **do**
11: $\quad\quad\quad \mathbf{r} \leftarrow \mathit{Row}(\mathcal{P} \setminus \mathit{Paths}(\mathcal{S}_l), \widehat{S})$
12: $\quad\quad\quad$ **if** $||\mathbf{rN}|| > 0$ **then**
13: $\quad\quad\quad\quad \widehat{\mathcal{P}} \leftarrow \widehat{\mathcal{P}} + \mathcal{P} \setminus \mathit{Paths}(\mathcal{S}_l)$
14: $\quad\quad\quad\quad \mathbf{N} \leftarrow \mathit{NullSpaceUpdate}\,(\mathbf{N}, \mathbf{r})$
15: $\quad\quad\quad\quad$ go to line 22

16: $\quad\quad\quad$ **for all** $e_j \in \mathcal{S}_l$ **do**
17: $\quad\quad\quad\quad \mathbf{r} \leftarrow \mathit{Row}(\mathcal{P} \setminus \mathit{Paths}(\{e_j\}), \widehat{S})$
18: $\quad\quad\quad\quad$ **if** $||\mathbf{rN}|| > 0$ **then**
19: $\quad\quad\quad\quad\quad \widehat{\mathcal{P}} \leftarrow \widehat{\mathcal{P}} + \mathcal{P} \setminus \mathit{Paths}(\{e_j\})$
20: $\quad\quad\quad\quad\quad \mathbf{N} \leftarrow \mathit{NullSpaceUpdate}\,(\mathbf{N}, \mathbf{r})$
21: $\quad\quad\quad\quad\quad$ go to line 22
22: **until** $\mathbf{N}$ has no columns left

23: **return** $\widehat{\mathcal{P}}$

Notation:
$\mathcal{A} \setminus \mathcal{B}$: subtract set $\mathcal{B}$ from set $\mathcal{A}$
$\widehat{\mathcal{P}} + \mathcal{P}$: add path set $\mathcal{P}$ to list of path sets $\widehat{\mathcal{P}}$

Appendix B

Notations

The notations used throughout this thesis are summarized in Table B.1. Chapters 4 and 5 have their own notations tables located at the end of the respective chapter.

The assumptions that we refer to throughout this thesis are summarized in Table B.2.

Notation	Definition
G	the network graph
E	the set of all links
V	the set of all nodes
V^H	the set of all hosts
V^R	the set of all routers
P	the set of all paths
$e_j \in E$	a link
$v_l \in V$	a node
$p_i \in P$	a path
$e_j \in p_i$	path p_i traverses link e_j
$\mathcal{E} \subseteq E$	a set of links
$\mathcal{P} \subseteq P$	a set of paths
$Links(\mathcal{P})$	all links traversed by paths in $\mathcal{P}$
$Paths(\mathcal{E})$	all paths that traverse links in $\mathcal{E}$
$\mathbf{R}$	the routing matrix
$Rank(\mathbf{R})$	the rank of the routing matrix
X_{e_j}	the logarithm of the transmission rate of link e_j
Y_{p_i}	the logarithm of the transmission rate of path p_i
Z_{e_j}	the congestion status of link e_j
W_{p_i}	the congestion status of path p_i
C	the set of all correlation sets
$\mathcal{C}_p \in C$	a correlation set
S	the set of all correlation subsets
$\mathcal{S}_k \in S$	a correlation subset
$\lvert A \rvert$	the number of elements in set A
$\widehat{\phi}_{e_j}(n)$	the transmission rate of link e_j during snapshot n
$\widehat{\phi}_{p_i}(n)$	the transmission rate of path p_i during snapshot n

Table B.1: Notations.

Assumption	Definition	Section
Routing Stability	The routing matrix does not change throughout the measurement period.	Section 2.1.1
Link Identifiability	All columns in the routing matrix are distinct.	Section 2.1.1
Stationarity	For any link e_j, the random variables $\widehat{\phi}_{e_j}(n)$, $n = 1, \ldots, N$, are identically distributed.	Section 2.1.2
Link Independence	The transmission rates of links, i.e., the random variables $\widehat{\phi}_{e_j}$, for all $e_j \in E$, are independent.	Section 2.2
Loss Uniformity	For any link e_j, the fraction of packets lost on link e_j is the same for all paths traversing the link.	Section 2.2
Separability	A path is good if and only if all the links it traverses are good.	Section 2.3
Probe Correlation	The network supports measurements that require perfect or strong temporal correlation between probes.	Section 2.5
Sparse Congestion	The percentage of congested links in the network is low.	Section 2.5
Link Homogeneity	All links are equally likely to be congested.	Section 2.5
No Fluttering Paths	Two paths never meet at one link, diverge, and then meet again at another link.	Section 3.1
Monotonicity of Link-Loss Variance	For any link e_j, the variance of X_{e_j} is a non-decreasing function of the corresponding link loss rate $1 - \widehat{\phi}_{e_j}$.	Section 3.1
Correlation Sets	Links are grouped into known correlation sets such that any two links belonging to different correlation sets are independent.	Section 4.2
Identifiability++	Any two correlation subsets are not traversed by the same paths.	Section 4.4.

Table B.2: Assumptions used by various network loss tomography algorithms.

Curriculum Vitae

Denisa Ghiţă

Education

2006 – 2012	Ph.D. in Computer Science École Polytechnique Fédérale de Lausanne (EPFL)
2001 – 2006	B.Sc. in Computer Science University "Politehnica" of Bucharest

Experience

2006 – 2011	Teaching Assistant at EPFL for: Advanced Computer Networks and Distributed Systems, TCP/IP, Informatique II, Stochastic Models for Communication Systems, and Middleware
Jun./Sep. 2007	Intern at Microsoft Research Cambridge, UK.
Mar./Sep. 2006	Intern at L3S Learning Lab Lower Saxony, Germany.

Publications

1. D. Ghita, H. Nguyen, M. Kurant, A. Argyraki, and P. Thiran, "Netscope: Practical Network Loss Tomography", *INFOCOM '10*
2. D. Ghita, K. Argyraki, and P. Thiran, "Network Tomography on Correlated Links", *IMC '10*
3. D. Ghita, C. Karakus, K. Argyraki, and P. Thiran, "Shifting Network Tomography Toward A Practical Goal", *CoNext '11*

Coordinates

Network Architecture Laboratory
School of Computer and Communication Sciences
École Polytechnique Féderale de Lausanne (EPFL)
CH-1015 Lausanne

Email: denisa.ghita@epfl.ch